# ADDRESSING THE DATA CHALLENGE IN AUTOMATIC DRUM TRANSCRIPTION WITH LABELED AND UNLABELED DATA

A Dissertation
Presented to
The Academic Faculty

By

Chih-Wei Wu

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Music

Georgia Institute of Technology

August 2018

**ADDRESSING THE DATA CHALLENGE IN AUTOMATIC DRUM TRANSCRIPTION WITH LABELED AND UNLABELED DATA**

Approved by:


Dr. Alexander Lerch, Advisor
School of Music
*Georgia Institute of Technology*

Dr. Mark A. Clements
School of Electrical and Computer Engineering
*Georgia Institute of Technology*

Dr. Grant Davidson
Dolby Research
*Dolby Laboratories, Inc.*

Dr. Jason Freeman
School of Music
*Georgia Institute of Technology*

Dr. Timothy Hsu
School of Music
*Georgia Institute of Technology*

Dr. Gil Weinberg
School of Music
*Georgia Institute of Technology*



Date Approved: May 8, 2018

## ACKNOWLEDGEMENTS

Foremost, I would like to express my gratitude to Alexander Lerch for his guidance, encouragement, and continuous support throughout my doctoral study. He led me into the field of music information retrieval and gave me the essential knowledge to explore this exciting research area. He is not only a great advisor, but also a mentor and a genuine friend to me. I am very proud of being one of the founding members of his research group at Georgia Tech, and I look forward to future collaborations.

I also want to thank all the committee members for their insightful comments and discussions. In particular, I would like to thank Mark Clements and Grant Davidson for their constructive feedback and unique perspectives. Additionally, I really appreciate Jason Freeman, Gil Weinberg, and Tim Hsu for their rigorous training through the coursework at GTCMT. I am truly grateful to have experts from both music technology and signal processing to provide their opinions for improving this work.

I was lucky enough to work with many people from the ISMIR community, and I would like to thank Christian Dittmar, Carl Southall, Richard Vogl, Gerhard Widmer, Jason Hockman, and Meinard Müller for the collaboration on the review article. This collaborative project is an important part of this thesis, and it has been an honor to work with the best researchers in the field and make contributions to the community.

Finally, I would like to dedicate this thesis to my family and my wife for their unconditional support. This work can not be finished without their love and sacrifice, and my gratefulness to them is beyond words.

## TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

## SUMMARY

Automatic Drum Transcription (ADT) is a sub-task of automatic music transcription that involves the conversion of drum-related audio events, such as drum onset times and playing techniques, into musical notations. While noticeable progress has been made in the past by combining pattern recognition methods with audio signal processing techniques, many systems are still limited by the difficulty of obtaining a meaningful amount of labeled data to support the data-driven algorithms; the lack of labeled data may lead to concerns such as the generality of the resulting models and the validity of the evaluation results. To address the challenge of insufficient labeled data in ADT, this work presents three approaches.

First, a dataset for ADT tasks is created. The creation process incorporates a semi-automatic process that minimizes the workload from human annotators, and the resulting dataset is verified both automatically and manually, ensuring the quality of the annotations.

Second, an ADT system that requires minimum labeled training data is designed. This system is based on a matrix factorization method specifically formulated to account for the presence of other instruments (e.g., non-percussive or pitched instruments). Additionally, the system adapts towards each individual signal with two template adaptation methods, providing flexibility and robustness in the case of unknown data.

Third, the possibility of improving generic ADT systems with a large amount of unlabeled data from online resources is explored. Specifically, two learning paradigms that are applicable to two major types of ADT systems are investigated. The first paradigm is *feature learning*, which learns appropriate feature representations from unlabeled data. The second paradigm is *student-teacher learning*, which transfers the expert knowledge from multiple teacher models to a student model through unlabeled data. Overall, this approach provides a scheme for data-driven ADT methods to leverage large unlabeled datasets and might have impact on other audio and music related tasks traditionally impeded by small amounts of labeled data.

# CHAPTER 1
# INTRODUCTION

Building an intelligent system that understands music is the ultimate goal of many researchers in the field of Music Information Retrieval (MIR). As pointed out by Schedl et al. [1], the extraction of meaningful information from music is one of the keys that allows effective indexing and content-based analysis in music. On a higher level, this process involves the conversion of audio signals to a semantic representation of music; a robust implementation of such a process would lead to the realization of "machine listening" for music. However, to achieve this goal, one might face several challenges. Particularly, the insufficient amount of music data with the corresponding labels (annotations) is one of the open-ended problems recognized by the MIR community.

In the following sections, this data challenge is introduced in the broader context of automatic music transcription. Additionally, its connection and implications to automatic drum transcription are presented. This chapter concludes with the motivation and research questions of this thesis.

## 1.1 Automatic Music Transcription

Automatic Music Transcription (AMT) is an active research area in MIR that concerns the conversion of audio signals into musical notation. Described as "a key enabling technology in music signal processing" by Benetos et al. [2], AMT aims to analyze the acoustical rendition of a musical idea, quantify the target events, and subsequently generate the representations that encapsulate this information. For example, to transcribe the pitch contour from the recording of a song, an AMT system needs to analyze the audio signal (i.e. waveform), segment and compute the pitch values, and return these values in other formats such as MIDI (Musical Instrument Digital Interface). A successful and accurate AMT system may

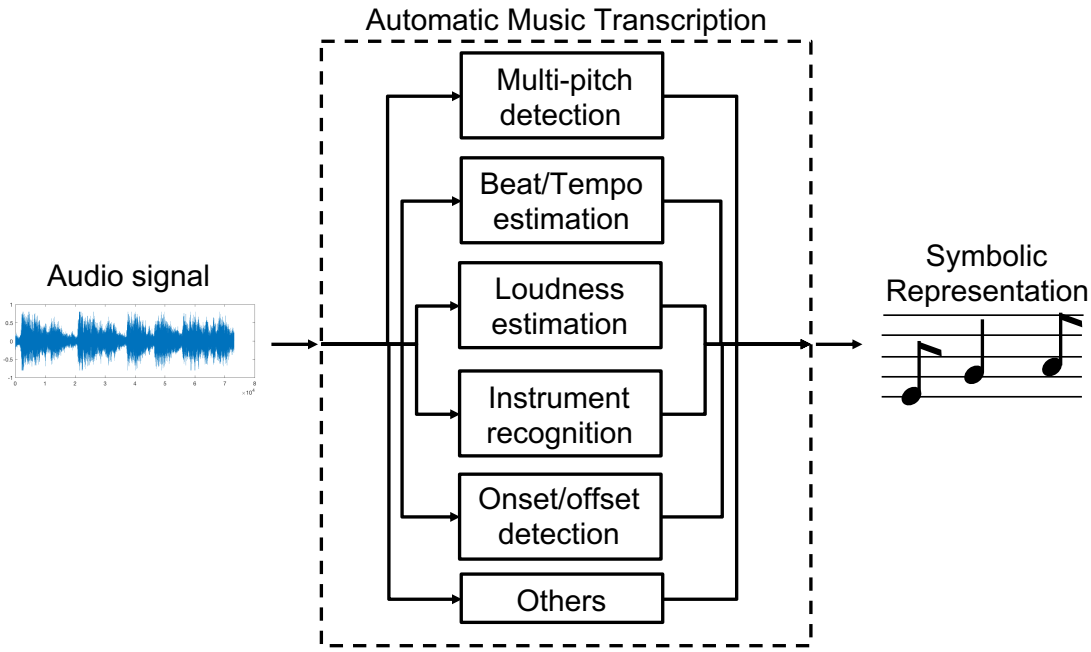


Figure 1.1: Illustration of the general process of automatic music transcription.

enable a variety of applications in fields such as music education and music production; furthermore, it can facilitate the documentation and study of specific music genres for which musical scores are not easily available (e.g., Jazz improvisation).

Generally speaking, AMT comprises different sub-tasks such as multi-pitch detection [3], onset detection [4], instrument recognition [5], and many others. As shown in Fig. 1.1, a complete AMT may require the integration of multiple systems in order to produce the symbolic representation of music (i.e., musical scores). Theoretically, these systems can either run in parallel or in series depending on the system design. In reality, many state-of-the-art systems for these sub-tasks still under-perform human experts [2], making the integration less reliable. As a result, many studies have been focusing on improving these sub-tasks. Automatic Drum Transcription (ADT), a task that extracts the drum related information from music, is one of these sub-tasks that is actively studied by the MIR community.

AMT systems, according to their decision making mechanisms, can be roughly categorized into: (i) rule-based systems and (ii) data-driven systems. The basic flowchart of

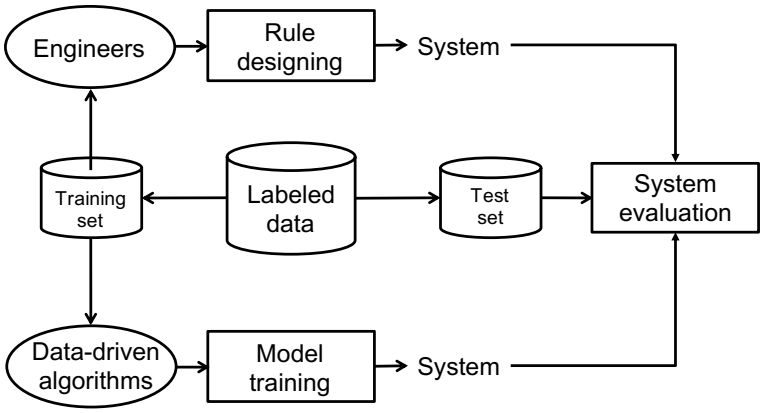


Figure 1.2: The process of building (top) Rule-based (bottom) Data-driven AMT systems.

these two types of systems is shown in Fig. 1.2. To start the process, a dataset with labels[1] is required. These labels represent the desired output (e.g., pitch values, onset locations, and instrument types) from the AMT systems given the corresponding data. The entire dataset is usually divided into two subsets for development and evaluation purposes. For *Rule-based* systems, the development starts with data observation/analysis by a human expert/engineer. The results lead to the manually designed rules that are based on domain knowledge and heuristics. In some cases, these rules are specific equations that compute the desired output directly. Once the rules are defined, the resulting system can be evaluated using the test set. The greatest advantage of this type of system is the interpretability. Since most of the rules are designed based on domain knowledge, the decision making mechanism tends to be transparent. In other words, a user who is knowledgable of the task should be able to adjust the parameters that are associated with the rules in order to optimize the performance under different scenarios. However, the downside of this approach is often its poor scalability. When the amount of data is too large, there could exist many edge cases that are undiscovered during the development phase. These cases are usually found only when the system fails, and new rules have to be manually designed accordingly.

*Data-driven* systems, compared to the *Rule-based* systems, require less domain knowledge and derive rules automatically. Given a training set and a selected data-driven algorithm,

[1] In this thesis, labels and annotations are used interchangeably

this type of system usually learns a function that maps the input to the desired output (labels) directly through a procedure known as "training". The resulting function (i.e., model) can be evaluated using the test set. On the one hand, this type of system does not rely on manually designed rules and generally scales better with more data. On the other hand, its interpretability tends to decrease when the model gets more complex. In certain situations, the parameters of the models might be abstract (e.g., due to multiple non-linear transformations), and the direct association between the parameters and the system output can be obscure.

Recently, more and more data-driven AMT systems report improved performance compared to the rule-based systems [6, 7]. As already shown in Fig. 1.2, labeled data is useful for building both types of AMT systems. With the increasing popularity of data-driven methods, the importance of labeled data becomes even higher. When the size of the labeled data is limited, two major concerns arise: (i) the model could easily overfit the data, which questions its generality, and (ii) the evaluation results could be overly optimistic due to the small sample pool size. To ensure the best outcome from the data-driven approach, "How much labeled data is needed?" is the critical question that one may have to consider. This question entails a complicated problem referred to as *Sample Size Determination* [8], and it is difficult to answer without heuristics and task-dependent insights. In the end, the availability of data is constantly listed as one of the top challenges in AMT [2] and general MIR research [1]. The difficulties pertaining to acquiring large amounts of labeled music data are introduced in the following section.

## 1.2 Availability of Labeled Music Data

The process of annotating music data is illustrated in Fig. 1.3, which consists of (i) a *data collection* step and (ii) a *data annotation* step. In the first step, a set of data that represents the research problem is collected. Next, the data is annotated by human experts with respect to the desired output for the specific task. For example, suitable data for ADT tasks would contain recordings that feature a variety of drum playings. Once collected, these data will

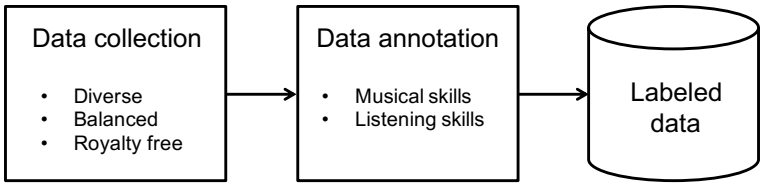


Figure 1.3: The general flow of creating labeled music data and the main considerations in each step.

be reviewed by the musicians who are proficient in transcribing drums and subsequently annotated with the drum types and their corresponding onset times.

The potential difficulties one might encounter in these steps are:

(i) the representativeness of the collected data: to build a dataset that represents the research problem well, the data has to be realistic, diverse, balanced, and royalty free. These criteria sometimes contradict each other, and the resulting data could thus be limited in certain aspects.

(ii) the consistency of the annotations: the annotation process requires both musical skills and listening skills from the annotators, and it is inherently subjective with respect to perceptual quantities such as pitch, timbre, and loudness [9]. Most importantly, the process is very labor-intensive and time-consuming.

In the case of AMT, which often requires note-level annotations such as pitch values, instrument types, and playing techniques, the above mentioned difficulties may increase; speeding up this process through crowd sourcing (e.g., Amazon Mechanical Turk[2]) is not feasible since it requires domain experts to complete the task. Similarly, gathering user submitted content from websites[3,4] may result in annotations with varying quality. Songle, a web-based interface for crowd sourcing music related annotations proposed by Goto et al. [10], tries to alleviate this problem by allowing the submission of user corrections, however, the overall consistency is still not guaranteed.

[2]https://www.mturk.com/mturk/welcome Last accessed: 2018/4/6
[3]https://www.ultimate-guitar.com, last access: 2018/04/07
[4]http://www.911tabs.com, last access: 2018/04/07

## 1.3 Motivation

In light of the above mentioned issues regarding the data availability in AMT, one may conclude that building and sharing large music databases is a non-trival task for the MIR community, and the availability of the labeled data has a strong influence on the research direction. For instance, Benetos et al. [2] pointed out that a large subset of AMT approaches only performed experiments on piano data, for which the audio aligned ground truth was easily obtained. This emphasis on piano may lead to models that are strongly biased towards piano-like instruments and cannot be generalized to other instruments. Likewise, ADT is also confined to the scope of the existing labeled datasets. In a project related to this thesis, which attempts to build an ADT system that detects various playing techniques, it was found that the number of occurences of these playing techniques were limited in the existing ADT datasets [11]. This sparsity of training data increases the difficulty for further advancing the performance of such systems.

Motivated by the current situation in AMT research concerning data, this thesis aims to address this challenge from three different angles: (i) contributing new annotated data for ADT tasks, (ii) designing algorithms that work under the constraint of limited resources (i.e., labeled data), and (iii) supporting data-driven systems by incorporating the nearly unlimited resources using unlabeled data. In particular, these concepts are applied to the problem of ADT. The goal is to not only further the progress of ADT under the data constraints, but also showcase the possibilities of improvement in the broader context of AMT and general MIR research.

A closer look at the ADT problem is the first step towards the embodiment of task-specific algorithmic designs. In Chapter 2, a comprehensive survey on ADT is presented to lay the groundwork for the later discussions. This includes the task definition, an overview of existing ADT approaches, and a summary of conventional evaluation metrics. In Chapter 3 – 5, the following research questions will be answered:

(i) **RQ1: How can the process of creating more labeled ADT datasets be improved?**

By examining the existing ADT datasets, several limitations can be observed. To address these issues, the simplest way is to create more labeled datasets with the desired properties. However, the creation of new datasets is difficult and time-consuming. What are the possible methods to reduce the cost of manual annotation and shorten the process? This thesis presents a collaborative effort that addresses this question in Chapter 3.

(ii) **RQ2: How to design an ADT algorithm that requires minimum amount of labeled data for training?**

ADT systems that require less labeled data to achieve a given level of performance are generally desirable, especially when the availability of data is challenged. How can one design an ADT system that requires minimum prior knowledge? How can such systems account for the variations of different signals and be generalizable? In this thesis, a signal adaptive ADT system that takes these considerations into account is described and evaluated in Chapter 4.

(iii) **RQ3: What are the most promising directions for general ADT systems to benefit from unlabeled data?**

Compared to the existing labeled data, unlabeled data have the advantage of including more diverse and realistic examples without the cost of human labeling. On the other hand, they could also be too noisy to be useful. Will those benefits outweigh the drawbacks? How can one select audio data to create a viable unlabeled dataset? What are the specific designs for different ADT systems to incorporate unlabeled data and how do they compare to each other? To this end, generic ADT systems are identified from the literature (see Sect. 2.3), and different methods for integrating unlabeled data are evaluated in an unified setting in Chapter 5. Additionally, the hypothesis on the usefulness of unlabeled data is examined statistically.

The answers to these questions lead to the conclusion of this thesis, which summarizes the possible directions for tackling the ADT problem under the data constraint and beyond in Chapter 6. Furthermore, the strong connection between ADT and the general AMT problems may enable the translation of the proposed methods from this thesis to other audio related tasks that are currently hindered by the availability of labeled data, encouraging the use of unlabeled data in the broader field of MIR research.

# CHAPTER 2
# AUTOMATIC DRUM TRANSCRIPTION

The content of this chapter has been prepared for the following manuscript:

- **Chih-Wei Wu, Christian Dittmar, Carl Southall, Richard Vogl, Gerhard Widmer, Jason Hockman, Meinard Müller, and Alexander Lerch, "A review of automatic drum transcription," IEEE/ACM Transactions on Audio, Speech and Language Processing, Vol. 26, No. 9, pp. 1457–1483, 2018.**

This is a joint work with several co-authors in the field of ADT. Every co-author contributes significantly to the success of this review article, and I would like to show my appreciation by acknowledging their efforts.

Following the introduction of the open challenges concerning the data availability in the field of AMT, this chapter focuses on the problem of ADT in more detail. As briefly introduced in Chapter 1, ADT involves the extraction of drum-related information from music signals. On the one hand, ADT systems focus on the detection and recognition of highly transient and impulsive events, which could be similar to other audio signal processing problems such as audio-surveillance [12] and acoustic event detection [13]. On the other hand, the musically organized drum events and the underlying vocabulary resemble human speech and language, which can be related to the well-studied fields such as speech recognition [14]. The combination of both makes ADT a unique research problem that might be of interest to the general audio signal processing community. In the next sections, the task definition, general approaches, and current challenges in the context of ADT are presented.

## 2.1 Introduction to Drum Kits

Drums, in the broadest definition, cover a wide spectrum of percussive instruments that are commonly used in Western and non-Western music (e.g., Tabla, Conga, and Timpani). Generally speaking, drums belong to a family of instruments called "membranophones" [15]; this type of instruments usually consists of a cylindrical body covered by a membrane. When the membrane (usually referred to as drum head) is struck by hand or sticks (e.g., drum sticks or mallet), it vibrates and resonates with the body, producing an impulsive sound with short decay time. In some publications, drums are also referred to as "unpitched percussive instruments" [16] in order to differentiate from pitched percussive instruments such as piano, vibraphone, or xylophone. Being one of the oldest musical instruments in history, drums are ubiquitous in many cultures and play an important role in emphasizing the rhythmic aspect of music. To increase the richness of the rhythmic patterns, drums are often played alongside various percussive instruments. The majority of these additional instruments belong to the family of "idiophones". This family of instruments features a rigid body of any shape and any material. When it is struck by a stick, the body vibrates as a whole, generating an inharmonic transient sound. When the idiophone is made of metal, its salient frequency is usually higher than typical drums, adding more colors to the musical palette.

In this thesis, the focus is on the drum kit, which is slightly different from the generic drums described above. A drum kit usually refers to a specific combination of percussive instruments that is well-known for its presence in Western music genres such as Pop, Rock, and Jazz. As shown in Fig. 2.1, a standard drum kit contains multiple pieces from both membranophones and idiophones. For example, the membranophones in a drum kit are Snare Drum (SD), Bass Drum (BD), High/Mid Toms (HT, MT), and Floor Tom (FT); the idiophones in a drum kit include HiHat (HH), Crash Cymbal (CC), and Ride Cymbal (RC). These instruments are typically arranged such that every piece is reachable when the

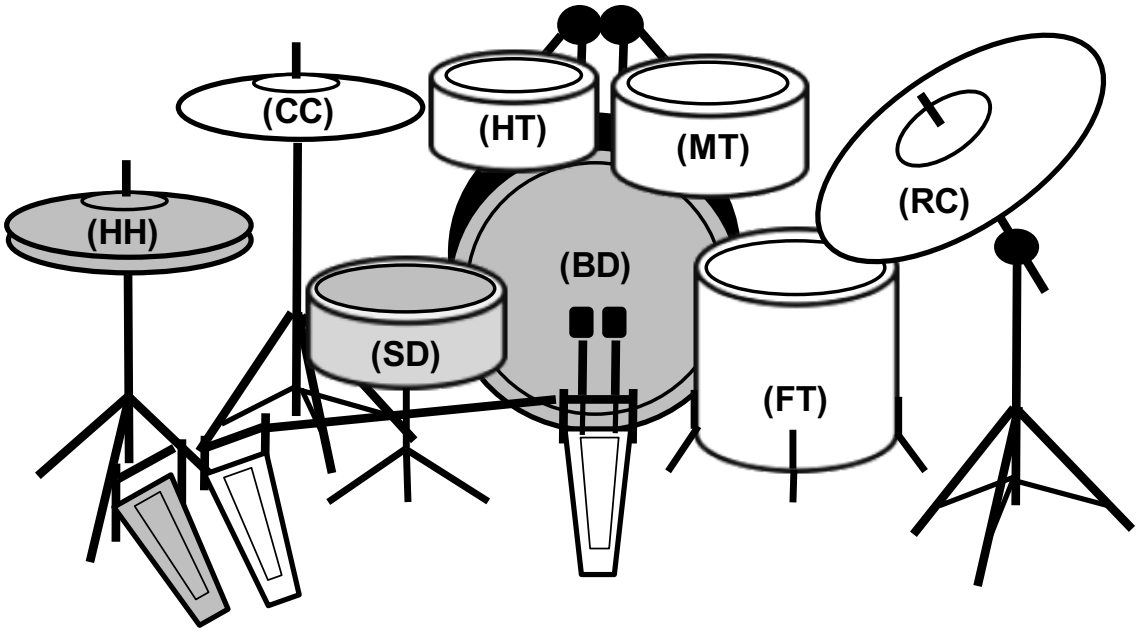


Figure 2.1: Illustration of a standard drum kit used in Western music. The instruments highlighted in gray color are HH, BD, and SD, which are the most essential components in many drum patterns of different musical styles.

drummer is sitting at the center. The number of instruments in a drum kit can be highly flexible, but at the core, it usually contains three most crucial instruments, namely the HH, BD, and SD (highlighted in gray color in Fig. 2.1). These instruments are the foundations of many rhythmic patterns, possibly due to their distinctive sound characteristics. The importance of these three instruments is also supported by the statistics of a typical drum dataset. For instance, in MDB Drums [17], the occurrences of HH, BD, and SD cover $85\%$ of the total number of drum events. As a result, many existing ADT studies only focus on these three instruments for their representativeness of a drum kit.

Figures 2.2 and 2.3 show the waveforms and log magnitude spectrograms of HH, BD, and SD. HH is an idiophone that consists of two crash cymbals and a stand with foot pedal; this foot pedal controls the gap between the two cymbals, which results in two different operating modes: closed and open. When HH is struck while closed (denoted as CHH), it produces a clicking sound that is highly transient. As shown in Fig. 2.2a, the waveform of CHH resembles an impulse with an exponential decay in amplitude, and the magnitude

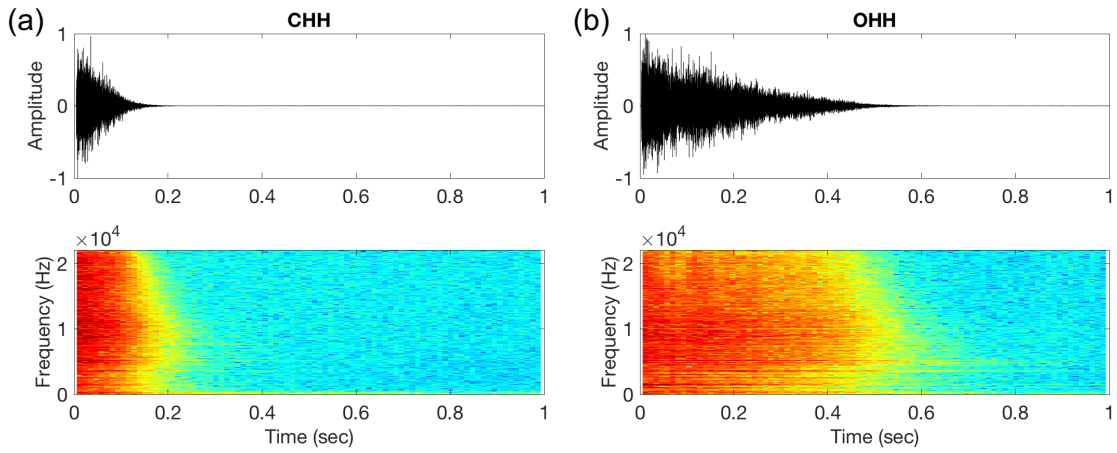


Figure 2.2: Waveform and magnitude spectrogram (frequency axis in log scale) of (a) CHH (b) OHH.

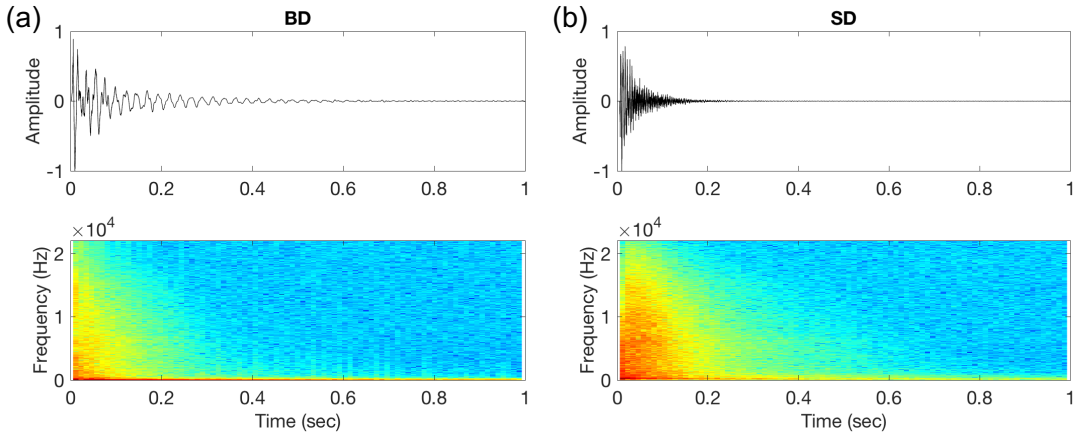


Figure 2.3: Waveform and magnitude spectrogram (frequency axis in log scale) of (a) BD (b) SD

spectrogram is similar to a broadband noise. When HH is struck while open (denoted as OHH), however, the produced noise burst tends to last longer. This can be observed from Fig. 2.2b, in which the waveform and the magnitude spectrogram of OHH are both elongated compared with CHH.

The construction of BD is straightforward; it is a larger membranophone with a foot pedal. The pedal is connected to a mallet, which will strike the membrane when it is triggered. This excitation creates a slow decaying sound with energies concentrated at the lower frequency region (as shown in Fig. 2.3a). SD is also a membranophone with an

additional snare belt attached under the lower membrane. When SD is struck with drum sticks, the vibration causes the snare belt to bounce against the lower membrane, creating a sizzling sound that is bright and fast decaying. As shown in Fig. 2.3b, the energies of the SD sound, compared with BD, are concentrated at a slightly higher frequency region. From Fig. 2.2 and Fig. 2.3, it is also clear that none of the sounds exhibit clear harmonic structure, which is very different from the pitched instruments. Overall, these sounds have a duration ranging from 50 ms to roughly 500 ms; the shorter sounds allow the consecutive drum events to be distinguished relatively easily.

In addition to the sounds triggered by simple strikes, most of the drum instruments can also produce sounds with timbral variations through different gestures. These gestures (or rudiments) are achieved by controlling the drum sticks in a specific way [18], and they are the foundations of many drum playing techniques. These rudiments can be categorized into four types:[1]

(i) Roll Rudiments: drum rolls created by single or multiple bounce strokes (Buzz Roll).

(ii) Paradiddle Rudiments: a mixture of alternative single and double strokes.

(iii) Flam Rudiments: drum hits with one preceding grace note.

(iv) Drag Rudiments: drum hits with two preceding grace notes created by double stroke.

There are also other playing techniques that are commonly used to create timbral variations in a drum kit, such as *Brush*, *Cross Stick*, and *Rim Shot*. In many cases, these techniques are used extensively on SD. For example, a *Roll* is achieved by pressing the drum sticks against the drum head in order to create multiple bounces in a short amount of time. As shown in Fig. 2.4a, this playing technique increases the duration of a single hit, giving a feeling of sustain to SD. A skillful drummer can generate fast bounces that blur the boundaries of these impulses, leading towards a more continuous sounding. A *Drag*

[1] http://vicfirth.com/40-essential-rudiments/ Last Access: 2018/04/11

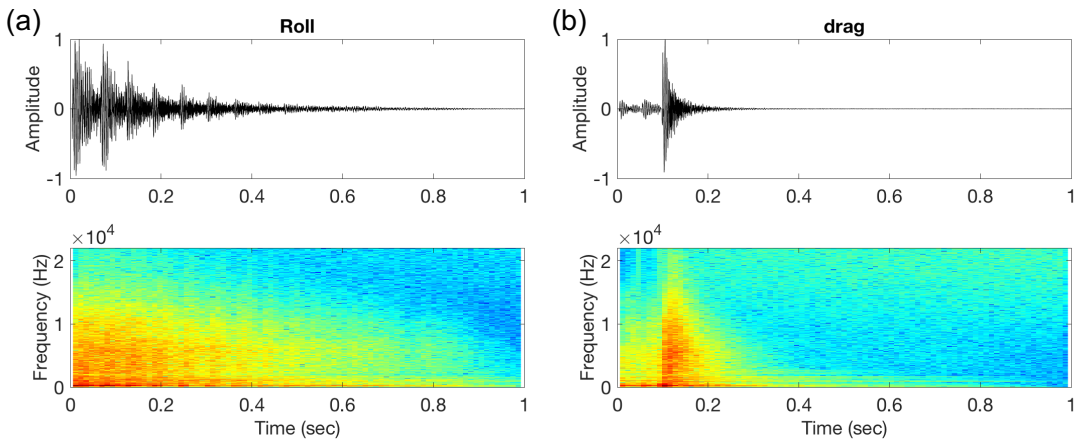


Figure 2.4: Waveform and magnitude spectrogram (frequency axis in log scale) of (a) Roll (b) Drag played on a SD

is another playing technique that requires precise control of the bounces prior to the main stroke. As shown in Fig. 2.4b, two spikes with lower amplitudes can be seen before the main peak. This technique allows the drummer to create a smoother transition between strokes and better articulate the rhythmic accents. Both examples demonstrate the possibilities of producing sounds with distinctive characteristics via different techniques. The difficulties of detecting these playing techniques will be discussed in the later sections (see Sect. 2.4.2). To simplify the problem while capturing the essence, most of the ADT studies only detect the basic strikes.

## 2.2 Task Definition

Following the description of the drum kit, this section presents the definition of the ADT problem and its related tasks. The general definition of ADT is similar to AMT (see Sect. 1.1) except for the focus on the drum instruments.[2] Simply put, ADT is a process that detects drum events from audio signals and subsequently converts them into other music notations (e.g., musical score and MIDI). In essence, this task relies on the robust recognition of the drum types and their onset times from audio streams; the conversion of this information

[2]In this thesis, the term "drum instrument" or "drum" is referring to the individual instrument within a standard drum kit.

into different formats would be relatively trivial if the information were correct. Therefore, the majority of existing ADT studies puts emphasis on improving the accuracy of detecting drum events. Depending on the target signals, ADT can be summarized as the following tasks:

(i) **Drum Sound Classification (DSC**): this is the most basic form of the ADT task, which involves the classification of isolated drum sounds. Each drum sound is a recording of a single drum hit, and the goal is to identify its source instrument as accurate as possible. This task is relatively straightforward, but it is an over-simplification of the real-world ADT problem.

(ii) **Drum Sound Similarity Search (DSSS)**: this task is similar to DSC, which operates on the isolated drum sounds. The goal of DSSS is to estimate the perceptual similarity between two isolated drum sounds. The resulting system can be used to retrieve drum sounds in a large database through the computed similarity.

(iii) **Drum Transcription on Drum only recordings (DTD)**: this task involves the transcription of drum events directly from a continuous audio stream that contains only drum sounds. As opposed to DSC and DSSS, which operate on isolated drum sounds, DTD requires an additional step to locate and segment these drum events. Furthermore, multiple drum sounds may occur simultaneously in the drum recordings, increasing the difficulty of this task. However, DTD makes less assumptions on the input data (e.g, pre-segmentation is not required) and is thus more generalizable to the real-world scenarios. This is one of the most well-studied ADT problems in the literature (see Sect. 2.3 for more discussions).

(iv) **Vocal Percussion Transcription (VPT)**: similar to DTD, this task also focuses on the transcription of drum events from audio streams. The only difference is the target signal; in VPT, the signal usually contains percussive-instrument-like sounds produced by a human artist through vocal techniques (often referred to as *beat boxing*).

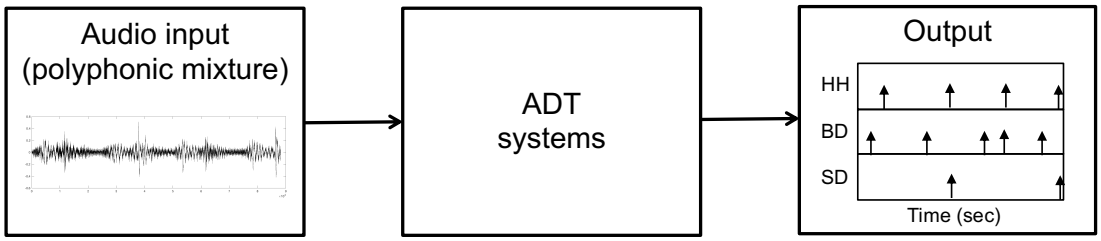


Figure 2.5: Illustration of the ADT task defined in this thesis.

Although some skillful beat boxers are capable of producing complex sounds (e.g., simultaneous drum sounds and various audio effects), the signals of beat boxing are usually monophonic, possibly due to the physical constraints of vocal tracts.

(v) **Drum Transcription in the presence of Melodic instruments (DTM)**: this task focuses on transcribing more complex signals. In DTM, the input signal is a polyphonic mixture that contains both drum and other melodic instruments (e.g., guitar, bass, and vocal), and the goal is to detect the drum events under the interference of these instruments. The additional instrumental sounds tend to overlap with the drums sounds in both time and spectral domain and increases the difficulty. This task is by far the hardest ADT problem, but it is also the most general formulation that is applicable to many real-world use cases.

(vi) **Drum Technique Classification (DTC)**: in addition to detecting the basic strikes, this task aims to recognize the playing techniques that are associated with each drum hit. Most of the existing studies define this task as a classification problem similar to DSC, and the evaluation is performed on the isolated recordings. In the real-world scenarios, however, a DTC system that works on polyphonic mixtures would be desirable. The research in this direction is currently hindered by the insufficiency of labeled data (see Sect. 3.2).

According to the definitions above, this thesis mainly focuses on the DTM task, for it represents the most generic scenario of ADT in the real-world applications. Specifically, the

expected output are the detected drum types and their onset times. The conversion of this information to musical scores is out of the scope of this work. This thesis revolves around three drum instruments, namely HH, BD, and SD. Similar settings can also be found in many existing ADT studies (see Sect. 2.3). To summarize, the definition of ADT in this thesis can be visualized as in Fig. 2.5.

## 2.3 General Approaches

In the following sections, an overview of the existing ADT approaches is presented. This includes the introduction of six generic *Design Patterns* that are identified from the previous studies. Based on these design patterns, existing ADT approaches can be categorized by their constituent building blocks.

### 2.3.1 Design Patterns

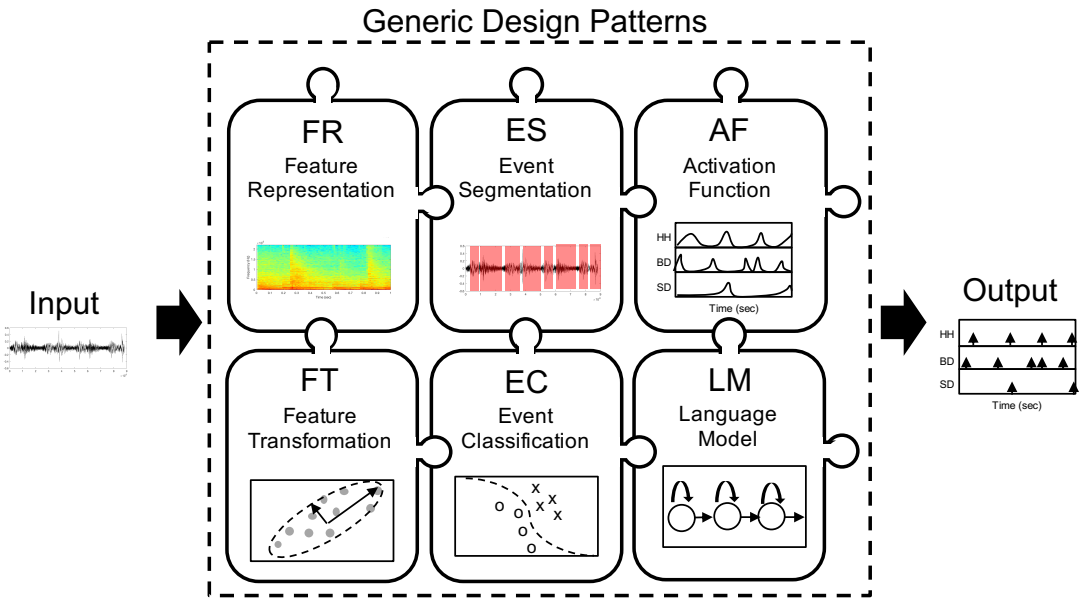


Figure 2.6: The proposed six generic design patterns that are relevant for ADT.

In earlier works on ADT, FitzGerald and Paulus [16] proposed to categorize the systems into two types, namely the pattern recognition and separation-based approaches. Later on, a

more refined grouping into four categories was proposed [19, 20]. These are:

(i) *Segment and Classify Approach*,

(ii) *Separate and Detect Approach*,

(iii) *Match and Adapt Approach*,

(iv) *HMM-based Recognition Approach*.

Considering the increasing amount of ADT research published, it became more and more difficult to draw clear boundaries between separate categories, and the traditional categorization might not accurately reflect the advances in ADT in recent years. As an alternative, a new paradigm is proposed as a collaborative effort to distinguish between methods according to their constituent building blocks [21]. Specifically, six generic design patterns that are used in most methods are identified as shown in Fig. 2.6.

These design patterns are building blocks to several ADT systems; they could be re-arranged and combined with no particular order. For most of the ADT systems, only a subset of these patterns are used. Note that the distinction between the proposed design patterns can sometimes be vague, and the specific algorithm for each pattern may vary depending on the ADT system. Additionally, these patterns are often not specific to drums, but rather inspired from related fields such as speech, language, and multimedia processing. For an introduction to the generic concepts and processing steps, please refer to [22, 23].
The proposed design patterns, compared to the traditional categorization, offer better flexibility in categorizing future ADT systems. Also, the modular way of analyzing existing approaches may contribute to the identification of un-explored or under explored combinations. In the following paragraphs, each one of these design patterns is introduced:

**Feature Representation (FR):** Apart from the time-domain waveform, discretized audio signals can also be converted into feature representations that are better suited for certain

processing tasks. A natural choice are Time-Frequency (TF) transforms (e.g., Short Time Fourier Transform, STFT), or Low-Level Features (LLF) derived from them. These representations are beneficial for untangling and emphasizing the important information hidden in the audio signal. This pattern also includes processing steps intended to emphasize the target drum signal in an audio mixture. These can either be based on spectral characteristics (e.g., band-pass filters, BPF, with predefined center frequencies and bandwidths) or based on TF characteristics (e.g., Harmonic-Percussive Source Separation, HPSS [24]).

**Event Segmentation (ES):** The main goal of this design pattern is to detect the temporal location of musical events in a continuous audio stream before applying further processing. This usually consists of computing suitable novelty functions (e.g., Spectral Flux) and identifying locations of abrupt change. A typical procedure would be to extract local extrema by applying a suitable peak-picking strategy, often referred to as onset detection (see Sect. 4.2.4 for more discussion) in MIR research.

**Activation Function (AF):** This design pattern seeks to map feature representations into activation functions, which indicate the activity level of different drum instruments. Different techniques such as NMF, Probabilistic Latent Component Analysis (PLCA) or Deep Neural Neworks (DNNs) are commonly used for deriving the activation functions.

**Feature Transformation (FT):** This design pattern provides a transformation of the feature representation to a more compact form. This goal can be achieved by different techniques such as feature selection, Principal Component Analysis (PCA), or Linear Discriminant Analysis (LDA). It should be mentioned that there is a strong overlap between the patterns **FT** and **AF** ; usually **FT** serves as a post-processing step for **FR** and arrives at a more compact feature representation, whereas **AF** is specifically used for converting the signal into drum-related activation functions. However, it should be noted that **AF** techniques can

also be used for **FT** purposes.

**Event Classification (EC):** This processing step aims at associating the instrument type (e.g., BD, SD, or HH) with the corresponding musical event. In the majority of papers, this is achieved through machine learning methods (e.g., Support Vector Machines, SVM) that can learn to discriminate the target drum instruments (or combinations thereof) based on training examples. Inexpensive alternatives include clustering (e.g., Alternate Level Clustering, ALC) and cross-correlation.

**Language Model (LM):** This pattern takes the sequential relationship between musical events into account. Usually this is achieved using a probabilistic model capable of learning the musical grammar and inferring the structure of musical events. **LM**s are based on classical methods such as Hidden Markov Models (HMM) or more recent methods such as RNNs.

The following sections will discuss various combinations and cascades of the introduced patterns in more detail. In each of the subsection headings, the typical cascade of patterns (e.g., **FR**, **ES**, **EC**) is given with the abbreviations introduced in Fig. 2.6. Note that these combinations are not exhaustive as new methods emerge constantly. However, with this flexible framework, it is possible to characterize future studies with different sets of cascaded patterns. Moreover, new design pattern can also be included, making this taxonomy extendable in the future.

### 2.3.2 Segmentation-based (FR, ES, EC)

This type of approach, as illustrated in Fig. 2.7, centers around the Event Segmentation **ES** concept and generally uses a cascade of Feature Representation **FR** and **ES** with occasional inclusion of Event Classification **EC**. Since most of the drum events are percussive

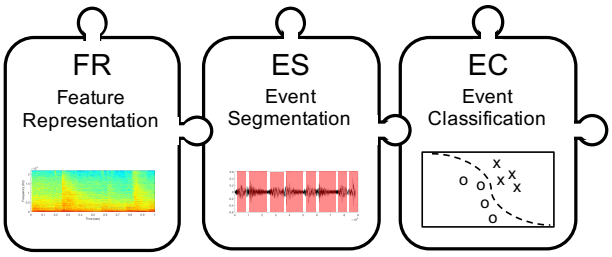


Figure 2.7: The combination of design patterns for the Segmentation-based approach.

and transient in nature, it is intuitive to apply a simple **ES** method (e.g., onset detection) on the input signal for segmenting and detecting such events. The rationale is to first emphasize the drum sound events within an audio mixture through various **FR** operations (e.g., HPSS, BPF), and perform **ES** on the resulting feature representations.

One of the earliest systems in this category was presented by Schloss [25]. The system estimates the envelope of the waveform and determines the attack with a threshold on the envelope-slope. Additionally, the decay time-constant is characterized by model fitting. By combining this information, the resulting system is able to detect basic strokes from drum-only recordings. Zils et al. [26] proposed a method starting with initial drum sound templates created from band-pass-filtered impulses. Next, the calculation of correlation between the time-domain signal and the initial templates, followed by a peak-quality assessment, is used as the event classification **EC** step. Finally, the templates are updated with the averaged time-domain signals of the detected events. This process is repeated until the number of detected events stops changing. While this *analysis by synthesis* approach has the advantage of requiring minimum prior knowledge, it has some potential issues due to its focus on time-domain signals, such as the confusion between high-pitched percussive sounds and singing voice, simultaneous events, and mismatches between initial template and the target drum sounds. These issues may become severe when the complexity of the audio mixture increases.

Another method of this category was proposed by Tzanetakis et al. [27]. The **FR** emphasises the characteristic frequency ranges of BD (30-280 Hz) and HH (2.7k-5.5k Hz) via

BPF based on the Discrete Wavelet Transform (DWT). Next, the **ES** and **EC** for each drum was done by onset detection on the extracted envelope of the time-domain sub-band signal. Since this method relies heavily on the selection of the frequency ranges of the filters, its generalization to other types of drum sounds can be problematic.

Kailakatsos-Papkostas et al. [28] proposed a similar method with a focus on real-time performance. First, multiple band-pass filters are applied followed by suitable amplifiers. Instead of using predefined frequency ranges, an iterative process is used to estimate optimal filter parameters (e.g., filter passband, stopband, onset detection threshold) by minimizing an objective function. Once the training is completed, a threshold is used to decide whether a drum is active or inactive. This method provides an alternative solution to the selection of characteristic frequency ranges of drums.

Generally speaking, the simplicity of the above mentioned methods has several advantages. First, the direct use of waveforms in the processing pipeline provides good interpretability of the results; this allows users with limited or minimal technical background to gain better control over the systems. Additionally, simple **FR** methods (such as BPF) and **EC** methods (such as cross-correlation or thresholding) can be implemented very efficiently, therefore enabling real-time applications, e.g., in the context of live music performances. However, such systems also have downsides. First, the robustness to additional sound components (e.g., coming from melodic instruments) might be insufficient. Since the systems typically use a simple **FR** step such as BPF to highlight the presence of drum events, they are susceptible to the interference of additional sounds. Second, these systems mainly use time-domain signals in favor of the fast processing speed. This potentially limits their capability of extracting more detailed information of the musical content, compared to other signal representations. Finally, the basic **EC** methods incorporated in this type of approach, such as cross-correlation and thresholding, might not be able to differentiate subtle timbral variations created by various playing techniques.

### 2.3.3 Classification-based (FR, ES, FT, EC)

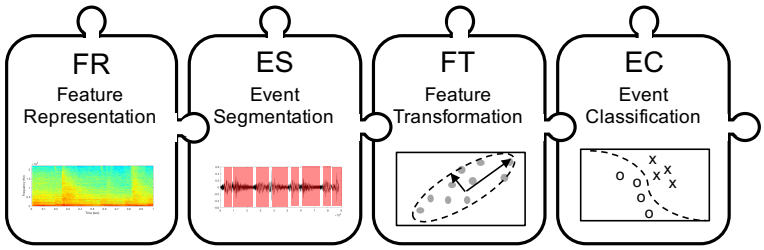


Figure 2.8: The combination of design patterns for Classification-based approach.

This type of approach builds around the Event Classification **EC** concept that differentiates different drum sounds using classifiers. The cascade of design patterns is shown in Fig. 2.8. *Classification-based* and *Segmentation-based* methods may look similar in terms of their cascaded patterns, but they are quite different in nature; *Segmentation-based* methods emphasize the efficiency and interpretability, whereas *Classification-based* methods focus on getting better performances with more sophisticated algorithms. There are many papers implementing this strategy; the basic idea is to extract Feature Representations **FR** from the audio signal, find the location of the potential events using Event Segmentation **ES**, refine the features with Feature Transformation **FT**, and then determine the instrument class of the events using Event Classification **EC**.

Since this processing pipeline is based on the standard pattern recognition paradigm, many different systems using different choices of **FR**, **FT**, and **EC** have been proposed. The most commonly used input representations are combinations of spectral features (e.g., centroid, flux, flatness), temporal features (e.g., zero crossing rate, local mean energy, RMS, envelope descriptors), and Mel-Frequency Cepstral Coefficients (MFCCs) [29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 19, 41, 42, 43]; other features, such as NMF derived features [44] and learned features [45], were also found useful in drum sound classification and drum transcription, respectively. To derive spectral features, mainly the STFT was used as **FR**; variants such as Constant-Q Transform (CQT) [40, 43], Line Spectral Frequencies (LSF) [46], and Mel-scale Log magnitude Spectrogram (MLS) [47]

have been shown to be viable options as well. Besides audio features, Gillet and Richard [48] proposed to use audio-visual features (AVF), which included features derived from video recordings of the drum performances. In contrast to the input representations, **FT** methods are optional and thus more situational. Techniques that were adopted in previous systems include Principal Component Analysis (PCA) [48], Information Gain Ratio [45], Recursive Feature Elimination [19], Correlation-based Feature Selection (CFS) [30] and Sparse Coding Matching Pursuit (SC-MP) [49, 50].

In terms of classifiers, basic models such as K-Nearest Neighbors (KNN) were often selected for their simplicity and interpretability [30, 31, 32, 35, 44, 41]. To account for non-linear relationships of the extracted features, SVMs with different kernel functions were used extensively in various systems [34, 35, 36, 37, 48, 39, 45, 19, 50, 51, 46, 52]; ensemble methods, such as Adaboost [34] and Random Forest (RF) [49], were often included in comparative studies for their effectiveness. Recently, successful models from other applications of machine learning, such as Convolutional Neural Networks (CNNs), have also been applied for drum sound classification [47]. In addition to the above mentioned supervised approaches, unsupervised methods were also applied for **EC**. For example, algorithms such as K-means [29, 40, 42] and ALC [43] were adopted to solve different ADT sub-tasks.

In Eronen's work on musical instrument recognition, a slightly different approach using a probabilistic model in the **EC** stage for classifying the drum sounds was presented [53]. Eronen proposed to use an HMM to model the temporal progression of features within an isolated audio sample. MFCC and the first derivative of MFCC were extracted as the features, followed by a **FT** step using Independent Component Analysis (ICA) that transforms the features into statistically independent representations.

Another system that falls implicitly into this category is the AdaMa-appraoch proposed by Yoshii et al. [54, 55, 56]. The general concept is to start with an initial guess for the drum sounds (sometimes called templates) that are iteratively refined to match the drum sounds

that actually occur in the target recording. The refinement is based on alternating between drum onset detection with the latest drum template estimate and updating the template with an averaged model of several, trustworthy onset instances of the drum sound. Unlike the system proposed by Zils et al. [26], AdaMa uses an STFT-based **FR** instead of raw waveforms, and an **EC** step based on a customized distance measure between the target event and the templates.

To summarize, the *Classification-based* methods have the following advantages. First, the general processing flow inherited from the pattern recognition paradigm allows an efficient and automated search of suitable settings. For instance, different classifiers or feature selection methods can be easily introduced in a modular fashion. Second, the possibility of adding various features during the **FR** step ensures the flexibility of incorporating expert knowledge in this type of system. However, since this type of system relies on a robust **ES** step to detect the musical events, any potential errors made in this stage are propagated through the system. Furthermore, to be able to handle simultaneous events (e.g., HH + SD, HH + BD), more classes are needed during the training phase. Thus, the number of class combinations will increase drastically as more instruments (e.g., HT, MT, LT, RC, and CC) are considered. Finally, *Classification-based* methods might have difficulties to recognize drum sound events in the presence of other melodic instruments that have never been presented to the system at training time, as the trained features are usually susceptible to the interference of the melodic instruments.

### 2.3.4 Language-model-based (FR, FT, LM)

Figure 2.9 shows the typical combination of design patterns for this type of approaches. After applying Feature Representation **FR** and Feature Transformation **FT** patterns, *Language-model-based* methods typically rely on a final processing stage, which involves the deployment of a Language Model **LM** to account for the temporal evolution of events on

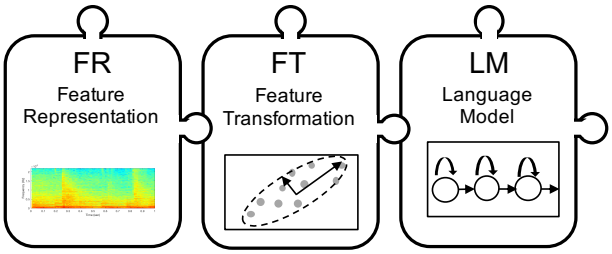


Figure 2.9: The combination of design patterns for Language-model-based approach.

a higher hierarchical level. Instead of detecting drum sound events directly from input representations, *Language-model-based* methods infer the underlying drum sound events by considering neighboring events and their probability as an entire sequence. This step is usually implemented using probabilistic models such as HMMs, where emission and transition probabilities are estimated from the temporal context of the training data.

One of the earliest works in this category was presented by Nakano et al. [57], which focused on VPT (i.e., beatboxing). The proposed system first extracts MFCCs from the given audio recording. Next, the acoustic features are decoded into sequences of onomatopoeic expressions using the Viterbi algorithm. Finally, the onomatopoeic expressions are mapped to drum sequences by retrieving the drum patterns with highest similarity from the predefined database. Another work that applies HMMs to model drum sequences was proposed by Paulus and Klapuri [58, 20]. In the **FR** step, the system uses a *sinusoids-plus-residual* model to suppress the harmonic components in the audio mixtures. Next, MFCCs are extracted as the feature representation, followed by a **FT** step using Linear Discriminant Analysis (LDA). Finally, the Viterbi algorithm and trained HMMs are used to determine the underlying drum sequences. Similarly, Şimşekli et al. [59] also use HMMs for detecting percussive events such as clapping and drum hits; with additional parameters, the model can be adjusted for the trade-off between accuracy and latency. Dzhambazov presented a HMM-based system that is aware of the bar positions [60]. The joint estimation of drum types and their corresponding bar positions allows the system to generate output that is compatible to symbolic representation. The authors report good performances on their

proprietary datasets, however, their generalizability on other datasets still needs to be further investigated.

In addition to decoding the underlying drum sequences, language models can also be used as post-processing tool. Gillet and Richard proposed to apply N-gram models on the symbolic data in order to fine-tune the detected onsets from the ADT systems in [61]. Their system first aligns the detected onsets to the tatum grid (a grid based on the smallest time unit inferred from the musical events). Next, the probability of a particular sequence can be estimated using a smoothed probability distribution of various sequences in the training corpus. Both supervised and unsupervised training schemes are evaluated, and the experiment results show a general performance gain of these methods. Nevertheless, the error from the preceding step (i.e., drum onset detection) may propagate through and reduce the overall performance.

The above mentioned methods are based on statistical estimation of the most likely drum sequence, and are hence aware of the musical context. In other words, these systems try to make predictions that are musically meaningful. For example, an unusual hit after certain sequences might be ignored due to the low probability of the resulting drum hit sequence. However, the main shortcoming of **LM** centered approach is the need of a large symbolic corpus, which is currently very limited (see Sect. 3.2 for more detailed discussions).

### 2.3.5 Activation-based (FR, AF, ES)

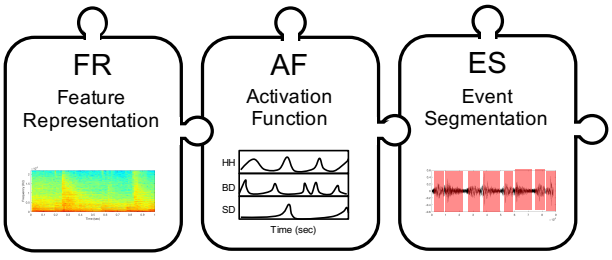


Figure 2.10: The combination of design patterns for Activation-based approach.

*Activation-based* systems, as shown in Fig. 2.10, often comprise a cascade of Feature Representation **FR**, Activation Function **AF**, and Event Segmentation **ES** steps. The defining factor of this approach is the concept **AF**, which generates the activity of a specific instrument over time. With the activation functions for every drum instrument, the **ES** step can be as simple as finding local maxima of those activation functions by means of suitable peak-picking algorithms.

There are basically two families of algorithms for deriving activation functions. The first one uses magnitude spectrograms as **FR** and applies matrix factorization algorithms as **AF** in order to decompose the spectrogram into basis functions and their corresponding activation functions. Early systems used methods such as Independent Subspace Analysis (ISA) [62], Prior Subspace Analysis (PSA) [63, 64, 65, 66], and Non-Negative Independent Component Analysis (NNICA) [67]. The basic assumption of these algorithms is that the target signal is a superposition of multiple, statistically independent sources. Even for drum-only recordings this assumption is problematic since the activations of the different drum instruments are usually rhythmically related. When the signal contains both drums and melodic instruments, this assumption may be more severely violated. Recently, more and more systems opted for NMF, which has less strict statistical assumptions about the sources. In NMF, the only constraint is the non-negativity of the sources, which is naturally given in magnitude spectograms. NMF-based ADT systems include basic NMF [68, 69] as well as related concepts such as Non-negative Vector Decomposition (NVD) [70, 71], Non-Negative Matrix Deconvolution (NMFD) [72, 73], Semi-Adaptive NMF [74], Partially-Fixed NMF [75, 76, 11], and PLCA [77]. Most of these factorization-based methods require a set of predefined basis functions as prior knowledge; when this predefined set does not match well with the components in the target signal, the resulting performance may decrease significantly.

The second family of algorithms which can be used to generate activation functions are based on Deep Neural Networks (DNN). In general, DNNs are a machine learning

architecture that allow to learn non-linear mappings of arbitrary inputs to target outputs based on training data. They are usually constructed as a cascade of layers consisting of learnable, linear weights and simple non-linear functions. The learning of the weight parameters is performed by variants of *gradient descent* [78]. In recent years, RNNs, a special form of DNNs designed to work on time series data, have been applied successfully for ADT. The use of bidirectional RNNs [79], RNNs with label time shift [80], as well as RNNs with Gated Recurrent Units (GRUs) and Long Short-Term Memory cells (LSTMs) [81, 82], showed comparable results to state-of-the-art systems. It is important to note that RNNs can in principle also perform sequence modeling, similar to the more classic methods such as HMM (see Sect. 2.3.4). However, this direction is still under-explored so far due to the lack of large training datasets (see Sect. 3.2 for more discussions). Recently, promising first attempts to apply CNNs and CRNNs to the task of ADT have been made [83, 82], showing the possibilities of adopting different architectures in addition to RNNs.

Overall, *Activation-based* methods have the advantage of producing intermediate output representations that are easy to interpret. Some of the factorization-based approaches can also be used to reconstruct the magnitude spectrogram of drum sources and serve as source separators. In addition, this type of approach takes care of simultaneous events without the need of introducing combined classes during training (see Sect. 2.3.3). However, when the multiple sources overlap in the spectral domain, cross-talk between activation functions will appear and degrade the performance. For instance, the activation function of a BD may also contain the interference from a bass guitar. Furthermore, the use of magnitude spectrograms neglects the phase, which could potentially strip away critical information.

A summary of all the reviewed ADT papers can be found in Table 2.1. This table provides essential information of each paper regarding its category, task, and constituent design patterns.

Table 2.1: A summary table of the existing ADT systems. *RT* means real-time systems.

| Year | Author(s) | Reference(s) | Task | Design Patterns | Category |
|---|---|---|---|---|---|
| 1985 | Schloss | [25] | DTD | **FR**, **ES** | *Segmentation-based* |
| 2000 | Gouyon et al. | [29] | DSC | **FR**, **EC** | *Classification-based* |
| 2002 | FitzGerald et al. | [62] | DTD | **FR**, **AF**, **ES** | *Activation-based* |
| 2002 | Herrera et al. | [30] | DSC | **FR**, **ES**, **FT**, **EC** | *Classification-based* |
| 2002 | Zils et al. | [26] | DTM | **FR**, **ES**, **EC** | *Segmentation-based* |
| 2003 | Eronen | [53] | DSC | **FR**, **ES**, **FT**, **EC** | *Classification-based* |
| 2003 | FitzGerald et al. | [63, 64, 65] | DTD | **FR**, **AF**, **ES** | *Activation-based* |
| 2003 | Herrera et al. | [31] | DSC | **FR**, **ES**, **FT**, **EC** | *Classification-based* |
| 2004 | Dittmar & Uhle | [67] | DTM | **FR**, **AF**, **ES** | *Activation-based* |
| 2004 | Gillet & Richard | [52] | DTD | **FR**, **ES**, **EC** | *Classification-based* |
| 2004 | Herrera et al. | [32] | DSC | **FR**, **ES**, **EC** | *Classification-based* |
| 2004 | Nakano et al. | [84] | VPT | **FR**, **LM** | *Language-model-based* |
| 2004 | Sandvold et al. | [33] | DTM | **FR**, **ES**, **FT**, **EC** | *Classification-based* |
| 2004 | Steelant et al. | [34, 36] | DSC | **FR**, **EC** | *Classification-based* |
| 2004 | Tindale et al. | [35] | DTC | **FR**, **ES**, **EC** | *Classification-based* |
| 2004 | Yoshii et al. | [54, 55, 56] | DTM | **FR**, **ES**, **EC** | *Classification-based* |
| 2005 | Degroeve et al. | [37] | DSC | **FR**, **EC** | *Classification-based* |
| 2005 | Gillet & Richard | [85] | DTM | **FR**, **ES**, **FT**, **EC** | *Classification-based* |
| 2005 | Gillet & Richard | [48] | DTM | **FR**, **ES**, **FT**, **EC** | *Classification-based* |
| 2005 | Hazan | [38] | VPT | **FR**, **ES**, **EC** | *Classification-based* |
| 2005 | Paulus & Virtanen | [68] | DTD | **FR**, **AF**, **ES** | *Activation-based* |
| 2005 | Tanghe et al. | [39] | DTM | **FR**, **ES**, **EC** | *Classification-based* |
| 2005 | Tzanetakis et al. | [27] | DTM | **FR**, **ES** | *Segmentation-based* |
| 2006 | Bello et al. | [40] | DTD | **FR**, **ES**, **EC** | *Classification-based* |
| 2007 | Gillet & Richard | [61] | DTM | **FR**, **ES**, **LM** | *Language-model-based* |
| 2007 | Moreau & Flexer | [44] | DTM | **FR**, **ES**, **EC** | *Classification-based* |
| 2007 | Roy et al. | [45] | DSC | **FR**, **ES**, **FT**, **EC** | *Classification-based* |
| 2008 | Gillet & Richard | [19] | DTM | **FR**, **ES**, **FT**, **EC** | *Classification-based* |
| 2008 | Pampalk et al. | [86] | DSSS | **FR**, **EC** | *Classification-based* |
| 2009 | Alves et al. | [69] | DTM | **FR**, **AF**, **ES** | *Activation-based* |
| 2009 | Paulus & Klapuri | [20, 58] | DTM | **FR**, **FT**, **LM** | *Language-model-based* |
| 2010 | Scholler & Purwins | [49] | DSC | **FR**, **EC** | *Classification-based* |
| 2010 | Spich & Zanoni | [66] | DTM | **FR**, **AF**, **ES** | *Activation-based* |
| 2011 | Şimşekli et al. | [59] | DTD | **FR**, **LM** | *Language-model-based* |
| 2012 | Battenberg | [71, 70] | DTD (RT) | **ES**, **FR**, **AF** | *Activation-based* |
| 2012 | Kaliakatsos et al. | [28] | DTD | **FR**, **ES** | *Segmentation-based* |
| 2012 | Lindsay-Smith et al. | [72] | DTD | **FR**, **AF**, **ES** | *Activation-based* |
| 2013 | Miron et al. | [41, 42] | DTD (RT) | **ES**, **FR**, **EC** | *Classification-based* |
| 2014 | Dzhambazov | [60] | DTM | **FR**, **LM** | *Language-model-based* |
| 2014 | Benetos et al. | [77] | DTM | **FR**, **AF**, **ES** | *Activation-based* |
| 2014 | Dittmar & Gärtner | [74] | DTD (RT) | **FR**, **AF**, **ES** | *Activation-based* |
| 2014 | Thompson & Mauch | [51] | DTM | **FR**, **ES**, **EC** | *Classification-based* |
| 2015 | Röbel et al. | [73] | DTM | **FR**, **AF**, **ES** | *Activation-based* |
| 2015 | Souza et al. | [46] | DSC, DTC | **ES**, **FR**, **EC** | *Classification-based* |
| 2015 | Rossignol et al. | [43] | DTM | **FR**, **EC**, **ES** | *Classification-based* |
| 2015 | Wu & Lerch | [75, 76] | DTD, DTM | **FR**, **AF**, **ES** | *Activation-based* |
| 2016 | Gajhede et al. | [47] | DSC | **ES**, **FR**, **EC** | *Classification-based* |
| 2016 | Vogl et al. | [80, 81] | DTD, DTM | **FR**, **AF**, **ES** | *Activation-based* |
| 2016 | Southall et al. | [79] | DTD, DTM | **FR**, **AF**, **ES** | *Activation-based* |
| 2016 | Wu & Lerch | [11] | DTC | **FR**, **AF**, **EC** | *Classification-based* |
| 2017 | Vogl et al. | [83] | DTM | **FR**, **AF**, **ES** | *Activation-based* |
| 2017 | Southall et al. | [82] | DTM | **FR**, **AF**, **ES** | *Activation-based* |
| 2017 | Wu & Lerch | [87] | DTM | **FR**, **AF**, **ES** | *Activation-based* |

### 2.3.6 Common Metrics

As discussed in Sect. 2.2, ADT studies cover a variety of tasks, and their evaluation metrics differ from each other. For tasks such as DSC and DTC, many previous studies (e.g., [29, 30, 35]) performed cross-validation on the collection of isolated drum sounds and reported the classification accuracy per instrument. For evaluating the accuracy, one may calculate *micro-averaged accuracy* and the *macro-averaged accuracy* [88] to account for the unevenly distributed and sparse classes. The metrics are defined in the following equations:

$$micro\ averaged = \frac{\sum_{k=1}^{K} C_k}{\sum_{k=1}^{K} N_k} \tag{2.1}$$

$$macro\ averaged = \frac{1}{K} \sum_{k=1}^{K} \left( \frac{C_k}{N_k} \right) \tag{2.2}$$

in which $K$ is the total number of classes, $N_k$ is the total number of samples in class $k$, and $C_k$ is the total number of correct predictions in class $k$. These two metrics have different meanings: while each sample is weighted equally for the micro-averaged accuracy, the macro-averaged accuracy applies equal weight to each class, which gives a better overview of the performance by placing more emphasis on the minority classes.

For tasks such as DTD and DTM, the main focus is to extract onset times of different drum instruments from a continuous audio stream. In this case, the metrics for assessing onset detection algorithms, namely Precision, Recall, and F-measure, are commonly used [58, 74, 76, 79, 80]. To compute these metrics, a tolerance window (e.g., 50 ms) must first be defined. A detected onset is counted as a *True Positive (TP)* if its deviation from the corresponding ground-truth annotation is less than the tolerance window. If a detected onset does not coincide with any annotated drum event, it is counted a *False Positive (FP)*; alternatively, if an annotated drum event does not coincide with any detected onset, it is counted as a *False Negative (FN)*. These three quantities define the standard Precision (P),

Recall (R), and F-measure (F) as shown in Eq. 2.3 to Eq. 2.5.

$$P = \frac{TP}{TP + FP} \tag{2.3}$$

$$R = \frac{TP}{TP + FN} \tag{2.4}$$

$$F = \frac{2 \cdot P \cdot R}{P + R} \tag{2.5}$$

The choice of tolerance window depends on many factors. According to Hirsh [89], the human perception of two separate audio events has a minimum gap as little as 2 ms. When the order of the events is taken into consideration, this gap becomes roughly 20 units. According to this measurement, a tolerance window up to 20 ms should be reasonable for ensuring perceptually similar reproduction. However, this narrow window is only meaningful when the events are precisely annotated in the ground-truth transcription. This is not a major issue when the dataset is synthetic (e.g., generated using MIDI). However, in real-world data the annotations are usually created manually by human annotators, and the averaged deviation tends to be higher than 20 ms due to human error. As a result, a larger tolerance window (e.g., 50 ms) is widely accepted in tasks such as ADT or onset detection [4]. Therefore, the tolerance window used in this thesis is 50 ms; this setting is consistent with the literature [19, 76, 79], although some authors use smaller tolerance windows such as 30 ms [58] and 20 ms [81] for the above described reasons.

## 2.4 Current Challenges

### 2.4.1 Interference of Multiple Instruments

The major challenge of state-of-the-art ADT systems usually comes from the interference of other instruments. The superposition of various instruments (e.g., guitar, keyboard, vocal,

or drums) makes the recognition of a specific instrument difficult due to overlaps in both spectral and temporal domain. Typically, the challenges arise in the presence of the following types of instruments:

**Percussive Instruments:** a basic drum kit, as introduced in Sect. 2.1, includes drums of different sizes and well-distinguishable timbral characteristics. However, in a more advanced setup for studio recordings, similar drums with subtle variations in timbre often appear, resulting in sounds that are hard to differentiate. This problem becomes more severe when these sounds occur simultaneously. In previous work, this problem is mostly addressed as a DSC task, in which the sounds are presented as isolated audio samples, and *Classification-based* methods tend to achieve a reasonably high classification accuracy. For example, a classification task for 33 different percussive instruments was performed by Herrera and Gouyon [31]; in the work done by Souza et al. [46], an attempt was made to classify different cymbals, such as china, crash, hi-hat, ride and splash. However, in a more realistic setting such as DTD, the perfect separation of simultaneous drum sound is hard to achieve. Thus, the classification accuracy can be expected to decrease compared with the DSC setting.

**Melodic Instruments:** despite fundamental differences between percussive and melodic instruments, the wide range of sounds produced from a drum kit can potentially coincide with sound components of many melodic instruments (e.g., the BD may overlap with bass guitar or SD may overlap with guitar and piano). As a result, DTM is considered much more challenging than DTD. Among all the methods in Table 2.1, only less than half of the systems were evaluated under the DTM setting, and most of them reported a noticeable drop in performance compared with DTD.

### 2.4.2 Playing Techniques

Playing techniques are an important aspect of expressive musical performances. For drum instruments, these techniques include basic rudiments (e.g., roll, paradiddle, drag, and flam) as well as timbral variations (e.g., ghost note, brush, cross stick, and rim shot). In spite of being an essential part of performances, most of the systems only focus on transcribing basic strikes, and the effects of different playing techniques are largely overlooked.

In an early attempt to transcribe playing techniques, Tindale et al. [35] presented a study on the automatic identification of timbral variations of the snare drum sounds induced by different excitations. A classification task is formulated to differentiate sounds from different striking locations (center, halfway, edge, etc.) with different excitations (strike, rim shot, and brush). Similarly, Prockup et al. [90] explored the discrepancies between expressive gestures on a larger dataset with combinations of different drums, stick heights, stroke intensities, strike positions, and articulations. In addition to membranophones, Souza et al. [46] thoroughly investigated different playing techniques for cymbal sounds. They differentiated either by the position where the cymbal is struck (bell, body, edge), how a hi-hat is played (closed, open, chick), or other special effects such as choking a cymbal with the playing hand. All of these studies showed promising results in classifying the isolated sounds, however, when the classifier is applied to the real-world recordings, the performance dropped drastically, as pointed out in [11]. Another attempt to retrieve playing techniques was proposed by Hochenbaum and Kapur through the use of both audio and accelerometer data [91]. However, the extra requirement of attaching the sensors to the performer's hands might impact the playing experience and deviate from the real playing gestures.

### 2.4.3 Recording Conditions and Post Production

Many of the existing ADT datasets feature audio examples that are recorded and produced in controlled environments. These data, while allowing easy manipulation of controlled

variables for experiments, often fall short in representing the diverse acoustic properties of the real-world drum recordings. In reality, drum recordings are likely to be convolutive, time-variant, and non-linear mixtures instead of simple superpositions of basic drum sounds. First, the room acoustics of the recording studio as well as the microphone setup might have substantial impact on acoustic properties such as reverberation. Next, different recording engineers have different preferences of applying equalization and filtering on the recorded signals. Additionally, non-linear audio effects, such as dynamic compression or distortion might be applied during the production. These factors would not be reflected if the dataset is constructed in a single, well-controlled environment.

The absence of these properties in the data can generally lead to the following problems: first, for data-driven-based approaches, the generality of the resulting system might become questionable. For example, if a system is trained on the clean audio signals only, its detection rate might deteriorate significantly when it is tested with noisy or reverberant signals. Second, for methods that are based on decomposition techniques, the general assumption of linearity might be violated. As a result, the performance of the systems will deteriorate significantly when test signal is heavily processed with non-linear effects. One possible strategy to address these challenges is through data augmentation. Applying different audio effects such as reverberation, distortion, and dynamic compression on training data could effectively increase the diversity of the dataset. However, the fundamental challenge of this approach is to augment the data in a musically meaningful way, which requires domain knowledge.

## 2.5 Summary

In this chapter, an overview of the existing ADT research is presented. This includes a general description of drum kits, the definition of various ADT tasks, and a new taxonomy

for categorizing ADT systems with the proposed generic design patterns. This chapter also discusses the current challenges in ADT research. Based on this discussion, the following trends can be concluded:

(i) Data-driven systems are prevailing: as summarized on Table 2.1, most of the existing ADT systems are data-driven. Specifically, both *Activation-based* and *Classification-based* methods are currently defining the state of the art.

(ii) DTM is still an open-ended research problem: many studies report a significant performance drop in DTM compared with DTD [19, 76, 79]. The potential causes of this phenomenon are discussed in Sect. 2.4.1, but a robust solution to this problem still remains undiscovered.

(iii) Systems that integrate language models are currently under-explored: as shown in Table 2.1, the majority of the ADT systems belongs to *Activation-based* methods, and the *Language-model-based* systems are still the minority. Although the concept of having models that are aware of the musical context is appealing, it usually requires a substantial amount of symbolic data for training. This scarcity of *Language-model-based* systems implies the need of more labeled data in ADT (see next chapter for more discussion).

(iv) Complete transcription is challenging: as discussed in Sect. 2.4.2, a complete drum transcription includes the detection of playing techniques. However, the discrimination between these techniques is difficult in the DTM setting. The ideal strategy to recognize these playing techniques is yet to be discovered. Additionally, a complete transcription requires the integration of other information such as time signature, tempo, and beat. Studies of this direction are rarely presented.

# CHAPTER 3

# DATASET FOR DRUM TRANSCRIPTION

The content of this chapter has been prepared for the following publication:

- **Carl Southall, Chih-Wei Wu, Alexander Lerch, and Jason Hockman, "MDB Drums – An Annotated Subset of MEDLEYDB for Automatic Drum Transcription," Late Breaking Demo of the International Society for Music Information Retrieval Conference (ISMIR), 2017.**

This dataset is a collaborative effort, and it should be noted that all authors contributed equally.

In addition to the combinations of design patterns as introduced in Chapter 2, the data used to train and evaluate ADT systems plays an important role in determining the success of the models. This chapter presents an overview of the popular ADT datasets that are publicly available; the insufficiency of these datasets are discussed in more detail, highlighting the data challenge in ADT. Furthermore, a collaborative project of creating a new labeled dataset for ADT is described. This project seeks to answer RQ1 in Sect. 1.3. As an intuitive approach to address the data insufficiency, this project incorporates a *Semi-automatic* process to generate labeled data in order to reduce the workload of human annotators. The details of this process and the statistics of the resulting dataset can be found in the later sections.

Table 3.1: An overview of the existing annotated datasets for ADT tasks. * indicates the dataset that is not freely available

| **Dataset** | **Suited for ADT task** | **Size (Duration)** | **Audio avail.** |
|---|---|---|---|
| 200 Drum Machines [92] | DSC | 7371 files (1 s each) | Y |
| Drum PT [11] | DTC | 30 files (30-90 s each) | N |
| DREANSS [93] | DTD/DTM | 22 files (10 s each) | N |
| ENST Drums [94] | DTD/DTM | 316 files (10-90 s each) | Y |
| IDMT-SMT-Drums [74] | DTD | 560 files (5-20 s each) | Y |
| MDB Drums [17] | DTD/DTM | 23 files (10-120 s each) | Y |
| MDLib2.2 [90] | DTC | 10624 files (1-2 s each) | Y |
| RWC-POP* [95] | DTM | 100 files (3-5 min each) | Y |
| Tindale et al. [35] | DTC | 1264 files(1 s each) | Y |

## 3.1 Existing Datasets

In Table 3.1, an overview of existing datasets is presented. These are often associated with different ADT tasks (see Sect. 2.2) and contain different types of recordings. For example, 200 Drum Machines [92] features a collection of electronic drum sounds, whereas as MDLib2.2 [90] only features acoustic drum sounds. As a result, the choice of dataset may have significant impact on the generalization capabilities of the resulting system. Among these publicly available datasets, IDMT-SMT-Drums [74] and ENST Drums [94] are two of the most commonly used datasets in recent ADT studies [74, 51, 73, 76, 80, 79, 83, 82, 87].

IDMT-SMT-Drums [74] comprises solely drum recordings containing the major drum instruments (i.e., HH, SD, BD). Each item in the dataset has a ground-truth transcription. There are 95 drum loop recordings from three drum kits (RealDrum, WaveDrum, and TechnoDrum). Most of the drum loops contain the basic drum patterns without applying special playing techniques; these drum patterns are commonly seen in Western music genres such as Pop and Rock. The sampling rate of every track is 44.1 kHz, and the total duration of the dataset is approximately two hours. In addition to drum loops, the dataset also includes several isolated drum hits for training. This dataset can be used for DSC and DTD tasks.

ENST Drums [94] comprises recordings of full drum kits, including instruments such

as CC, RC, HT, MT, and LT (as shown in Fig. 2.1). Again, each item in the dataset has a corresponding ground-truth transcription available. These recordings are played by three different drummers on their own drum kits. Each set of recordings from each drummer contains individual hits, short phrases of drum beats, drum solos, and short excerpts played with accompaniments (i.e., the *minus one subset*). These accompaniments are prepared as separated files, allowing the remixing with the corresponding drum tracks. Additionally, these accompaniments feature both acoustic instrument sounds (e.g., bass, guitar, saxophone, clarinet) and the synthetic sounds (e.g., synthesized using MIDI). All accompaniments are temporally aligned to the drum recordings. Playing techniques such as ghost notes, flam, and drag are present in many recordings without ground-truth annotations.[1] The sampling rate of each recording is 44.1 kHz. This dataset can be used for DTD and DTM tasks.

The above mentioned datasets, while being limited in certain aspects (see Sect. 3.2 for a detailed discussion), provide a great starting point for most ADT tasks. Therefore, both of the datasets are currently considered as benchmark datasets for ADT research.

## 3.2 Insufficiency of Existing Datasets

As summarized in Table 2.1, many of the existing ADT systems are based on data-driven machine learning approaches. However, with the complexity of music, the difficulty of generating labels, and the restrictions of intellectual property laws, building and sharing annotated datasets becomes a non-trivial task; many of the commonly used datasets are thus limited in different aspects. A closer look at the existing datasets shown in Table 3.1 reveals the following limitations:

**Size:** The most common issue of all the existing drum transcription datasets is the insufficient amount of data. Overall, the datasets that contain only audio samples with single

[1] The annotations are available as part of the contributions of this thesis [11].

drum hits (Tindale et al. [35], 200 Drum Machines [92], and MDLib2.2 [90]) have more files, whereas the datasets that contain entire drum sequences (ENST Drums [94], IDMT-SMT-Drums [74], DREANSS [93], RWC-POP [95] and Drum PT [11]) have less files. However, the total duration of each dataset is usually less than a few hours and might not be representative for the wide variety of real-world music. A small dataset could also increase the risk of being overfit by the models. Furthermore, since these datasets are created under very different conditions, they cannot be easily integrated into one large entity. The small data size is especially detrimental for *Language-model-based* systems, which usually require a large amount of symbolic data for training. This could be one of the reasons why *Language-model-based* systems are curretly less popular than the other types of approaches.

**Complexity:** The existing datasets have the tendency to over-simplify the ADT problem. For example, in datasets containing isolated drum hits (i.e., Tindale et al. [35], 200 Drum Machines [92], and MDLib2.2 [90]), the transcription problem is reduced to the classification of different drum sounds; in IDMT-SMT-Drums [74], only drum sequences with basic patterns are presented in the dataset, and the subtle gestures such as playing techniques are missing. The lack of complexity results in datasets that are unrealistic for real-world use cases.

**Diversity:** Most of these datasets do not cover a wide range of music genre and playing style. For instance, RWC-POP [95] only covers Japanese-pop music, IDMT-SMT-Drums [74] only covers basic patterns and playing techniques for pop and rock music, and ENST Drums [94] only features playing styles from 3 drummers. The limitation in terms of diversity can hinder the system's capability of analyzing a wider range of music pieces. Particularly, the lack of any singing voice in the corpora ENST Drums and IDMT-SMT-Drums indicates their potential insufficiency.

**Homogeneity:** The problem of homogeneity usually originates from the creation of the dataset. Since each dataset is most likely to be generated under fixed conditions (i.e., recorded in the same room by the same group of performers), the audio files within the same dataset tend to have high similarities. This is very different from real-world scenarios, where the drum recordings come from different musicians, different drum kits, and different recording and processing conditions (as discussed in Sect.2.4.3). This limitation in homogeneity can potentially lead to an overfitting issue in the resulting ADT systems.

## 3.3 MDB-Drums Dataset

Based on the discussion in the previous section, one may conclude that the field of ADT is in need of datasets which address the aforementioned limitations. To that end, the most straightforward solution is to create more datasets with the desirable attributes (e.g., more real-world music with diversity and complexity). However, as discussed in Sect. 1.2, the process of creating annotations is both skill-demanding and time-consuming. In an attempt to tackle these issues, MDB Drums [17] was created collaboratively. In particular, a semi-automatic process is incorporated, reducing the workload and ensuring the quality of the resulting dataset. More details are presented in the following sections.

### 3.3.1 Overview

The main idea behind the creation of MDB Drums is to explore an alternative solution to reduce the workload of creating annotations through a semi-automatic process. Such a process leverages relatively robust MIR systems (e.g., onset detection, see Sect. 4.2.4) to minimize the effort from human annotators. As shown in Fig. 3.1, this process starts by selecting a suitable dataset. The goal of this project is to create a realistic drum dataset without introducing the extra cost of recording and post-processing, and MedleyDB dataset [96] is chosen as it contains multi-track recordings, allowing for easier annotation. Specifically, the *MusicDelta* subset is selected for its diversity in music genres and the reasonable duration

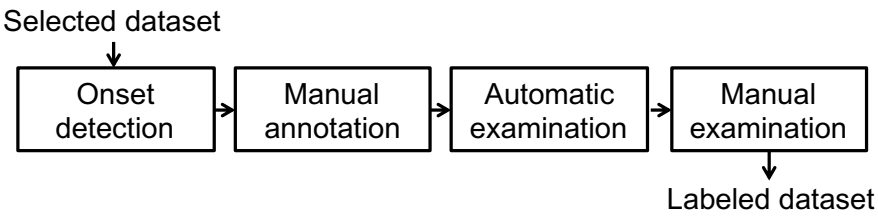


Figure 3.1: The flowchart of semi-automatic process for annotating MDB drums [17].

of each track. Next, an onset detector is used to annotate the onset locations of the drum hits automatically. Since the drum track is isolated from the mixture, a reasonably robust output can be expected from this automated step. Once the onsets are located, the human annotator only need to label each onset with its corresponding drum type. To control the quality of the annotations, automatic checks on the typical errors are implemented, followed by the manual examination/correction as the final step. These steps are elaborated as follows.

### 3.3.2 Annotation Process

Dataset creation begins with the annotation stage, which consists of two steps: (i) onset detection and (ii) manual annotation. To facilitate the process, the OnsetDetector algorithm from the Madmom library [97] is applied to each of the audio files. One example file from the dataset is shown in Fig. 3.2. Comared to the polyphonic mixture, the drum only recording provides a cleaner representation of the percussive events. Therefore, the use of the drum-only tracks in MedleyDB allows the onset detector to consistently achieve reliable results. Next, the extracted onset times are imported into Sonic Visualizer[2] for refinement by human annotators. Each onset is annotated with its corresponding drum instrument name and playing technique, and the missing onsets are added.

### 3.3.3 Examination Process

The examination stage also consists of two steps: (i) automatic examination and (ii) manual examination. These two steps are implemented to ensure the consistency and the accuracy

[2]http://www.sonicvisualiser.org, last access: 2018/04/22

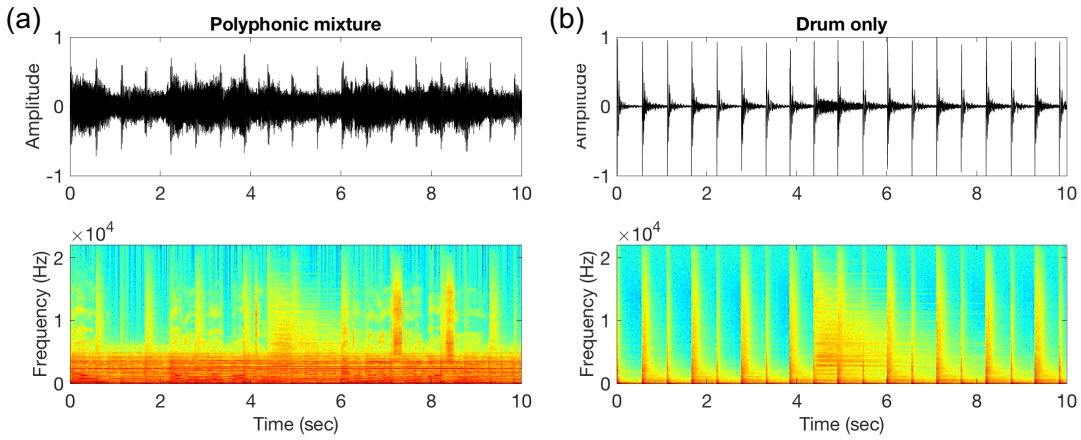


Figure 3.2: An example from the MDB Drums dataset that shows the waveform and magnitude spectrogram (frequency axis in log scale) of its (a) polyphonic mixture and (b) drum only recording.

of the resulting annotations. In the automatic examination step, three types of errors are automatically checked: (i) invalid drum instrument name (i.e., incorrect labels), (ii) duplicate labels within a 50 ms window, and (iii) three or more different labels within a 50 ms window. These rules are defined based on the heuristics and domain knowledge of the annotators. These automatic checks highlight the problematic labels, and a manual examination is subsequently performed to fix these errors. The manual examination follows a "cross-annotator" validation procedure which requires the annotators to validate each other's assignments. During the process, approximately 180 onsets (roughtly 2%) have been corrected. This estimation indicates the relatively high agreement between the annotators. Finally, the dataset is examined by an external reviewer for further verification.

### 3.3.4 Dataset Details

The *MDB Drums* dataset consists of a total of 23 tracks with an average length of 54 seconds. As the dataset contains multi-track files (i.e., all the isolated instrumental tracks are included), a variety of combinations can be easily generated (e.g., drum + guitar, drum + bass guitar).

There are 7994 onsets in the entire dataset, and a detailed list of these onsets is shown in

Table 3.2: An overview of the onset numbers in MDB Drums. Similar abbreviations as in Fig. 2.1 are used with the following additions: Tom Tom (TT), Cymbals (CY), and Other Percussion (OT)

| Instrument Type | Included Playing Techniques or Variants | #Onsets |
|---|---|---|
| BD | N/A | 1539 |
| SD | Brush, Drag, Flam, Ghost note, No snare | 2654 |
| HH | Close hihat, Open hihat, Pedal hihat | 2639 |
| TT | High tom, High-mid tom, High floor tom, Low floor tom | 90 |
| CY | Ride cymbal, Ride cymbal bell, Crash cymbal, China, Splash | 1002 |
| OT | Side stick, Tambourine | 70 |

Table 3.2. The dataset is available in the Github repository.[3]

## 3.4 Conclusion

This chapter presents an overview of the existing ADT datasets. The examination of these datasets highlights the data challenge in ADT and implies the need for additional labeled datasets. To address this challenge, a collaborative effort of creating a new ADT dataset is described. As opposed to the traditional approach of manual annotation, the proposed approach incorporates a semi-automatic process that reduces the workload of human annotators and improves the annotation efficiency. Additionally, both the automatic and manual examination are implemented to ensure the quality of the resulting dataset.

The potential downside of this approach is the scalability. Since it requires a multi-track dataset in order to get reliable output from the onset detector, it is hard to generalize the same procedure to any arbitrary music data. Also, the annotation process still requires human involvement despite the use of an onset detection algorithm; the correctness of this process relies on the skills of the human annotators and their interpretation of ambiguous sounds, and this task cannot be easily assigned to people with no specific experiences through crowd-sourcing. In addition, even with the assistance of an onset detector, the averaged time for annotating one track is approximately 1–2 hrs. As a result, this approach alone is not sufficient to address the data challenge in ADT.

[3] https://github.com/CarlSouthall/MDBDrums, last access: 2018/04/22

# CHAPTER 4
# DRUM TRANSCRIPTION WITH LIMITED DATA

The content of this chapter has been published in the following publications:

- **Chih-Wei Wu and Alexander Lerch, "Drum Transcription using Partially Fixed Non-Negative Matrix Factorization," Proceedings of the European Signal Processing Conference (EUSIPCO), 2015.**

- **Chih-Wei Wu and Alexander Lerch, "Drum Transcription using Partially Fixed Non-Negative Matrix Factorization With Template Adaptation," Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), 2015.**

In an attempt to answer RQ2, this chapter describes an ADT system that is designed for use cases when the training data is limited. This system can be categorized as *Activation-based* according to the taxonomy introduced in Chapter 2. It applies a matrix-factorization-based algorithm to derive activation functions and uses a predefined drum dictionary as prior knowledge. The construction of this dictionary only requires a minimum amount of training examples, which is ideal for the real-world applications. Furthermore, the proposed algorithm adapts the predefined drum dictionary during testing, providing a flexible scheme to alleviate the potential problem of template mismatching. In the next sections, the details of this system, including the signal model, algorithm description, implementation, and the evaluation results, will be presented.

## 4.1 Introduction

In Chapter 2 and 3, the general approaches in ADT and the challenging situation with the data availability were described. As concluded in Sect. 2.5, data-driven systems have shown the most promising results for various ADT tasks. However, with the limited amount of labeled data, splitting the dataset for training and testing purposes is necessary and has profound impact on the results. While a larger split of training data is needed for the algorithms to learn from more examples, reserving more data for testing gives a more reliable estimation of the system's performance. As a result of this dilemma, the generality of the data-driven systems is often one of the main concerns. To that end, an ADT system that is less demanding on the training data would be desirable under the data constraints.

Among all types of ADT systems reviewed in Sect. 2.3, *Segmentation-based* methods generally require the least amount of training data due to their simplicity in the design patterns. For example, some of the *Segmentation-based* systems use a simple combination of band-pass filtering and peak-finding to detect the presence of different drums [27, 28]. By carefully choosing the frequency range for each band-pass filter, this type of system can work reasonably well without requiring a large amount of training data. However, the optimal choice of the filter parameters could be situational; the best settings might vary with the signals, and the fine-tuning could be difficult to execute manually. Additionally, this simple method tends to fail in the case of DTM, where multiple instruments could fall into the same frequency band as the target drums.

Another type that could potentially minimize the use of training data are the *Activation-based* approaches with matrix factorization methods. As discussed in Sect. 2.3.5, this type of approach requires a set of predefined basis functions (i.e., dictionary) as the prior knowledge, and it usually works well in the DTD setting. The dictionary can be constructed by extracting templates from the example drum sounds; in some cases, only one example from each drum instrument is sufficient for reasonable performance. Based on these considerations, this

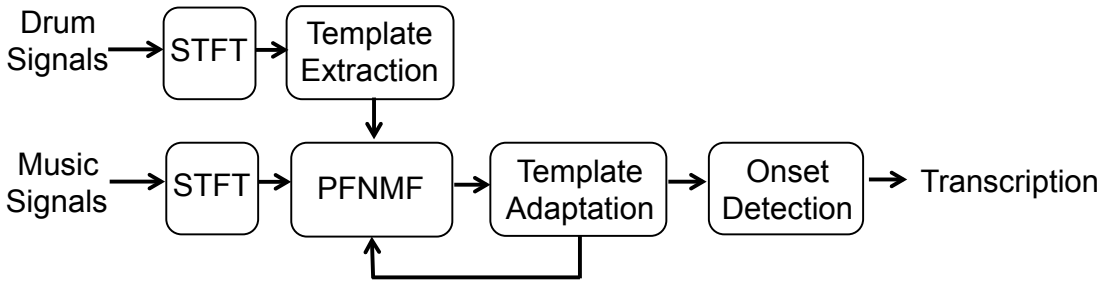


Figure 4.1: Flowchart of the implemented drum transcription system using PFNMF.

thesis explores the similar idea of using matrix factorization methods with predefined dictionaries. Since the goal is DTM, the proposed system is designed specifically to reduce the interference of other melodic instruments. More details are given in the following sections.

## 4.2 Method

### 4.2.1 Overview

An overview of the proposed ADT system is shown in Fig. 4.1. This system consists of two phases, namely the dictionary preparation and transcription phase. In the dictionary preparation phase, the process starts by calculating the input representations (i.e., STFT) from the audio signals. These signals include the isolated drums sounds of the targeted drum types (i.e., HH, SD, BD). Next, a set of drum templates is extracted from the drum signals. These templates are subsequently used to construct the drum dictionary matrix.

In the transcription phase, the same procedure is applied to compute the input representations from the test signals. These input representations are further decomposed using Partially-Fixed Nonnegative Matrix Factorization (PFNMF) with the predefined drum dictionary (see Sect. 4.2.2). To account for the mismatches between the dictionary and the actual drum sounds within the test signals, two signal adaptive methods are applied to iteratively update the predefined drum dictionary (see Sect. 4.2.3). Once the optimization process is completed, the system returns an activation matrix which contains the activation functions

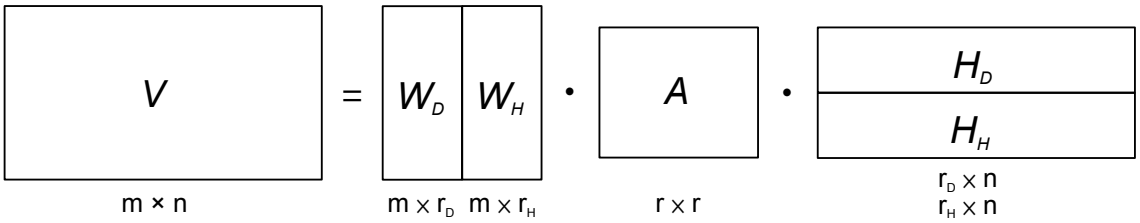


Figure 4.2: Illustration of the factorization process. $W$: dictionary matrix, $H$: activation matrix; Subscript $_{\mathrm{D}}$: drum, subscript $_{\mathrm{H}}$: harmonic components. $\mathrm{A}$ is the weighting matrix.

that correspond to the activity levels of different drums. Finally, a simple peak-picking process is applied on all activation functions to determine the location of the drum hits. More details of the above mentioned steps are provided in the following sections.

### 4.2.2 PFNMF

The basic concept of NMF can be expressed as $V \approx WH$ with non-negativity constraints, in which $V$ is a $m \times n$ matrix, $W$ is a $m \times r$ dictionary matrix, and $H$ is a $r \times n$ activation matrix, with $r$ being the rank of the NMF decomposition. In most audio applications, $V$ is the magnitude spectrogram with $m$ frequency bins and $n$ frames, $W$ contains the magnitude spectra of the salient components, and $H$ indicates the activation of these components with respect to time [98]. The matrices $W$ and $H$ are estimated through an iterative process that minimizes a distance measure between the target spectrogram $V$ and its approximation [99].

The idea of using NMF with prior knowledge of the target source within the mixture has been applied to source separation tasks [100] and multipitch analysis [101]. However, to adjust this algorithm for transcribing drum events in polyphonic mixtures, the following aspects should be considered: (i) the method should be applicable to real-world scenarios in which users only have limited amount of training samples that are slightly different from the target source, (ii) the method should be able to adapt to different content in polyphonic mixtures, and (iii) the method should be both efficient and easily interpretable.

Based on these considerations, PFNMF [75] was proposed as a signal adaptive method for ADT. Inspired by the source separation method proposed by Yoo et al. [102], the concept

of PFNMF can be visualized as in Fig. 4.2: the matrices $W$ and $H$ are split into the matrices $W_{\mathrm{D}}$ and $W_{\mathrm{H}}$, and $H_{\mathrm{D}}$ and $H_{\mathrm{H}}$, respectively. The algorithm initializes the matrix $W_{\mathrm{D}}$ with drum templates and does not modify it during the factorization process. The matrices $W_{\mathrm{H}}$, $H_{\mathrm{H}}$, and $H_{\mathrm{D}}$ are initialized randomly. The rank $r_{\mathrm{D}}$ of $W_{\mathrm{D}}$ and $H_{\mathrm{D}}$ depends on the number of templates (i.e., percussive instruments) provided, and the rank $r_{\mathrm{H}}$ can be arbitrarily chosen.

By increasing the rank $r_{\mathrm{H}}$, a larger $W_{\mathrm{H}}$ will be initialized to better adapt to the target signal, however, this unbalanced increase in templates would also decrease the weight of the drum templates in the optimization process, thus reducing the impact of the percussive templates on the NMF cost function. This effect is reduced by the weighting matrix $\mathrm{A}$ which balances the weights between drum and harmonic templates.

The total rank $r = r_{\mathrm{D}} + r_{\mathrm{H}}$. $A$ is a $r \times r$ diagonal weighting matrix, which contains weighting coefficients for every template to balance the drum and harmonic dictionaries in the NMF cost function. In this thesis, the coefficients are set to be $\alpha$ and $\beta$ for drum templates and harmonic templates, respectively. They can be expressed as the following equations:

$$\alpha = \frac{(r_{\mathrm{D}} + r_{\mathrm{H}})}{r_{\mathrm{D}}}, \tag{4.1}$$

$$\beta = \frac{r_{\mathrm{H}}}{(r_{\mathrm{D}} + r_{\mathrm{H}})}. \tag{4.2}$$

This setting is to increase the weighting of drum templates and slightly decrease the weighting of harmonic templates as $r_{\mathrm{H}}$ becomes larger. When $r_{\mathrm{H}} = 0$, the algorithm reduces to the original NMF.

The choice of cost function (i.e., distance measure) and the multiplicative update rules follow the similar configuration as described by Lee and Seung [99]. The distance measure

used is the generalized KL-divergence:

$$D_{\mathrm{KL}}(X \mid Y) = \sum \left( X \odot \log \left( \frac{X}{Y} \right) - X + Y \right), \tag{4.3}$$

in which $X$ and $Y$ are two matrices, $\odot$ is the element-wise multiplication, and the division is also element-wise. The NMF cost function as shown in Eq. 4.4 is minimized by applying gradient decent and multiplicative update rules.

$$J = D_{\mathrm{KL}}(V \mid \alpha W_{\mathrm{D}} H_{\mathrm{D}} + \beta W_{\mathrm{H}} H_{\mathrm{H}}) \tag{4.4}$$

The matrices $W_{\mathrm{H}}$, $H_{\mathrm{H}}$, and $H_{\mathrm{D}}$ will be updated according to Eqs. (4.5)–(4.7):

$$H_{\mathrm{D}} \leftarrow H_{\mathrm{D}} \odot \frac{W_{\mathrm{D}}^T (V/(\alpha W_{\mathrm{D}} H_{\mathrm{D}} + \beta W_{\mathrm{H}} H_{\mathrm{H}}))}{W_{\mathrm{D}}^T} \tag{4.5}$$

$$W_{\mathrm{H}} \leftarrow W_{\mathrm{H}} \odot \frac{(V/(\alpha W_{\mathrm{D}} H_{\mathrm{D}} + \beta W_{\mathrm{H}} H_{\mathrm{H}})) H_{\mathrm{H}}^T}{H_{\mathrm{H}}^T} \tag{4.6}$$

$$H_{\mathrm{H}} \leftarrow H_{\mathrm{H}} \odot \frac{W_{\mathrm{H}}^T (V/(\alpha W_{\mathrm{D}} H_{\mathrm{D}} + \beta W_{\mathrm{H}} H_{\mathrm{H}}))}{W_{\mathrm{H}}^T} \tag{4.7}$$

Finally, the presented method before template adaptation can be described as the following steps:

(1) Construct a $m \times r_{\mathrm{D}}$ dictionary matrix $W_{\mathrm{D}}$, with $r_{\mathrm{D}}$ being the number of drum components to be detected.

(2) Given a predefined rank $r_{\mathrm{H}}$, initialize a $m \times r_{\mathrm{H}}$ matrix $W_{\mathrm{H}}$, a $r_{\mathrm{D}} \times n$ matrix $H_{\mathrm{D}}$ and a $r_{\mathrm{H}} \times n$ matrix $H_{\mathrm{H}}$.

(3) Normalize $W_{\mathrm{D}}$ and $W_{\mathrm{H}}$.

(4) Update $H_{\mathrm{D}}$, $W_{\mathrm{H}}$, and $H_{\mathrm{H}}$ using Eqs. (4.5)–(4.7).

(5) Calculate the cost of the current iteration using Eq. (4.4).

(6) Repeat step 3 to step 5 until convergence (the error between two consecutive iterations changes by less than 0.1% or the number of iterations exceeds 300).

The time positions of the drum events can then be extracted by applying a simple onset detection on the rows of matrix $H_{\mathrm{D}}$.

### 4.2.3 Template Adaptation

Using template adaptation in drum transcription process can be found in previous studies. These approaches usually start with seed templates and gradually adapt them to the optimal templates [56, 74]. Based on the similar concept, two signal adaptive methods for template adaptation with PFNMF are proposed. Both methods have the same criterion to stop iterating when the error between two consecutive iterations changes by less than $0.1\%$ or the number of iterations exceeds 20. However, the adaptation process typically converges after 5–10 iterations.

*Adaptation Method 1 (AM1): Complementary Update*

In the first method (referred to as AM1), the drum dictionary $W_{\mathrm{D}}$ is updated based on the cross-correlation between the activations $H_{\mathrm{H}}$ and of each individual drum in $H_{\mathrm{D}}$. PFNMF starts by randomly initializing a $W_{\mathrm{H}}$ with rank $r_{\mathrm{H}}$. Although $W_{\mathrm{H}}$ tends to adapt to the harmonic content, it may still contain entries that belong to percussive instruments due to a spectral shape mismatch between the initialized drum templates and the target sources. This will result in cross-talk (simultaneous activation) between $H_{\mathrm{H}}$ and $H_{\mathrm{D}}$ and generate two similar-shaped activation functions both with lower amplitude. However, these harmonic templates may also provide complementary information to the original drum templates. To identify these entries, the normalized cross-correlation between $H_{\mathrm{H}}$ and $H_{\mathrm{D}}$ for each individual drum is computed using Eq. (4.8)

$$\rho_{x,y} = \frac{\sum_{j=1}^{n} x(j) \cdot y(j)}{\|x\|_2 \cdot \|y\|_2}, \tag{4.8}$$

where $x$ and $y$ represent different activation vectors, and $n$ is the number of samples in the activation vectors. A threshold $\rho_{thres}$ is defined for identification of related entries, and the drum template $W_{\mathrm{D}}$ can be updated using Eq. (4.9), where

$$W'_{\mathrm{D}} = (1-\gamma)W_{\mathrm{D}} + \gamma\frac{1}{S}\sum_{i=1}^{S}\left(\rho^{(i)}W_{\mathrm{H}}^{(i)}\right) \tag{4.9}$$

$$W_{\mathrm{H}}^{(i)}(i = 1, ..., S) \tag{4.10}$$

are the entries with their corresponding $\rho_{x,y}$ higher than $\rho_{thres}$, and $S$ is the number of the selected entries. Since a low $\rho_{thres}$ can introduce too much adaptation and vice versa, a threshold of $\rho_{thres}$ = 0.5 is chosen heuristically. The amount of adaptation also depends on the weighting factor $\gamma = \frac{1}{2^k}$, which decreases as iteration number $k$ increases.

*Adaptation Method 2 (AM2): Alternate Update*

In the second method (referred to as AM2), the drum template $W_{\mathrm{D}}$ is adapted by alternatively fixing $W_{\mathrm{D}}$ and $H_{\mathrm{D}}$ during the decomposition process. The adaptation process starts by fixing $W_{\mathrm{D}}$, and PFNMF will try to fit the best activation $H_{\mathrm{D}}$ to approximate the drum part in the music. Once $H_{\mathrm{D}}$ is determined, a new iteration of PFNMF can be started by fixing $H_{\mathrm{D}}$ and allow $W_{\mathrm{D}}$, $W_{\mathrm{H}}$ and $H_{\mathrm{H}}$ to update. This constraint will guide the algorithm to fit better drum templates based on the detected activation $H_{\mathrm{D}}$. The update rule for $W_{\mathrm{D}}$ is shown in Eq. (4.11).

$$W_{\mathrm{D}} \leftarrow W_{\mathrm{D}} \odot \frac{(V/(\alpha W_{\mathrm{D}}H_{\mathrm{D}} + \beta W_{\mathrm{H}}H_{\mathrm{H}}))H_{\mathrm{D}}^T}{H_{\mathrm{D}}^T} \tag{4.11}$$

Note that both AM1 and AM2 are based on ad hoc updates and the convergence is not always guaranteed. However, in practice, these methods generally converge to better solutions within a reasonable number of iterations (see Sect. 4.3.3).

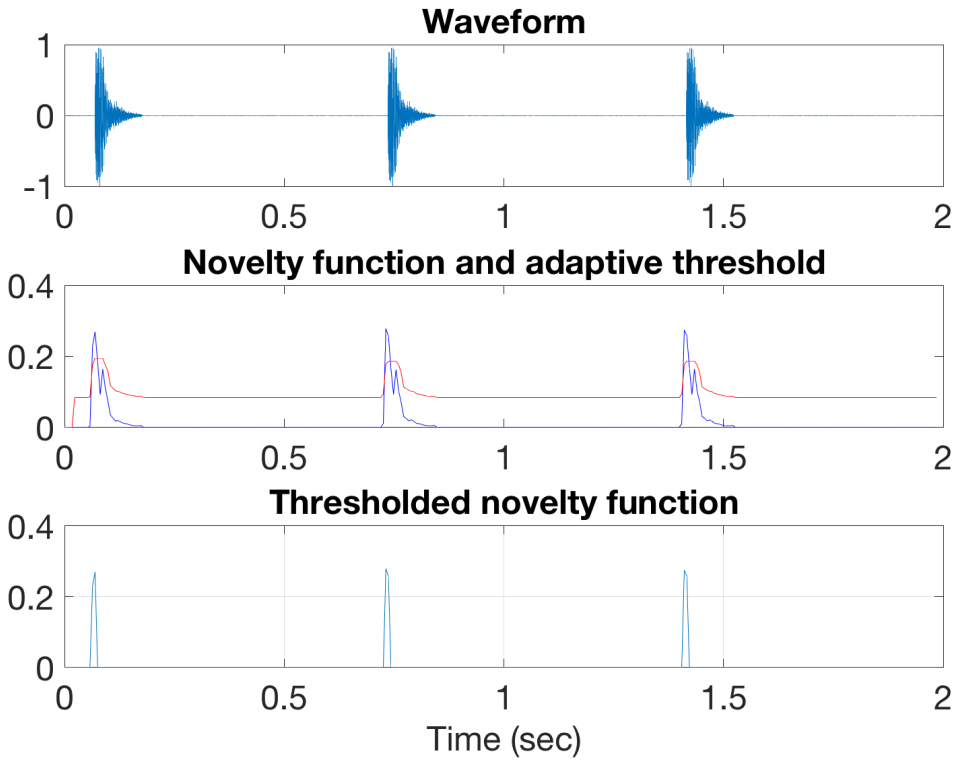


Figure 4.3: Example of the basic onset detection process (top) original waveform (middle) computed novelty function and the adaptive threshold; the threshold is marked in red color (bottom) the novelty function after applying the threshold

### 4.2.4 Onset Detection

In the context of MIR research, an onset is usually referring to the starting point of a musical event in the audio signal. To be more specific, an onset, according to Bello et al. [4], is the starting point of a *transient* state (i.e., a short time interval in which the signal is quickly evolving). The general process for onset detection, as described by Lerch [22], includes the following two steps:

(i) **Novelty function**: this step computes a continuous function that describes the amount of audio changes over time. In other words, this function represents the amount of "new" information in the audio signal at any given time step.

(ii) **Peak picking**: this step identifies the local maxima from the novelty function, which

is filtered by subtracting an adaptive threshold. These local maxima are the onset locations of the signal.

A simple example of onset detection is shown in Fig. 4.3. Given the waveform of an audio signal, a novelty function using *Spectral Flux* is first computed. This novelty function $f$ can be defined as:

$$f(n) = \frac{\sqrt{\sum_{m=0}^{M}(V(m,n) - V(m,n-1))^2}}{M}, \tag{4.12}$$

in which $V$ is the magnitude spectrogram of the input signal, $m$ is the frequency bin index, $M$ is the total number of frequency bins, and $n$ is the block index.

Next, a signal adaptive threshold can be computed. The purpose of this threshold is to suppress the noise in the novelty function and reduce the number of *False Positives* (see Sect. 2.3.6) of the onset detection process. In this example, the adaptive threshold $t$ is computed using a median filter, which can be expressed as:

$$t(n) = \lambda * max(f) + median(f(n), p), \tag{4.13}$$

in which $f$ is the novelty function, $\lambda$ is the offset coefficient relative to the maximum value, $p$ is the order (length) of the median filter, and the $n$ is the block index.

Once the adaptive threshold is computed, the novelty function can be thresholded by setting every value below the adaptive threshold to zero. The thresholded novelty function, as shown in Fig. 4.3, shows clear spikes that correspond to the onsets. Finally, these onsets can be detected by finding the local maxima of this thresholded novelty function.

The example shown above is a simple demonstration of the process. Onset detection is a relatively mature task in MIR research, and more advanced novelty functions and peak picking techniques have been proposed and proven successful on various kinds of audio

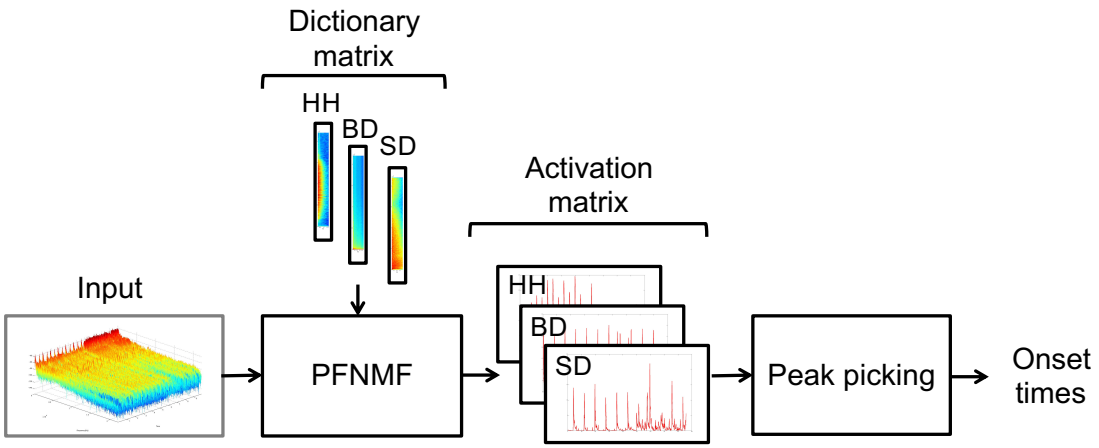


Figure 4.4: Flowchart of the process for detecting drum onsets times from the activation functions.

signals. A comprehensive introduction to these techniques can be found in [4, 22].

From the previous example, it can be easily observed that a novelty function resembles an activation function from NMF. Therefore, a similar process as described above is applied to pinpoint the exact locations of drum hits. As shown in Fig. 4.4, PFNMF decomposes the input magnitude spectrogram and returns an activation matrix that contains the activation functions of every drum instrument. To locate the drum hits, these activation functions can be treated as novelty functions directly, and the standard procedure for peak picking, including the adaptive thresholding, may be applied to transcribe the onset times.

### 4.2.5 Implementation

The main input representation to PFNMF is the STFT of the signals, which is calculated using a Hann window with a window length and a hop size of $2048$ and $512$ samples, respectively. The resulting magnitude spectrogram is used as the input representation. All signals are resampled to a sampling rate of 44.1 kHz and down-mixed to mono prior to the computation of STFT.

A predefined dictionary matrix $W_{\mathrm{D}}$ is constructed from the training set, which consists of isolated drum sounds. This training set is a subset of the ENST dataset, which contains audio tracks of 5 to 6 single hits for each drum, performed by three drummers. For every

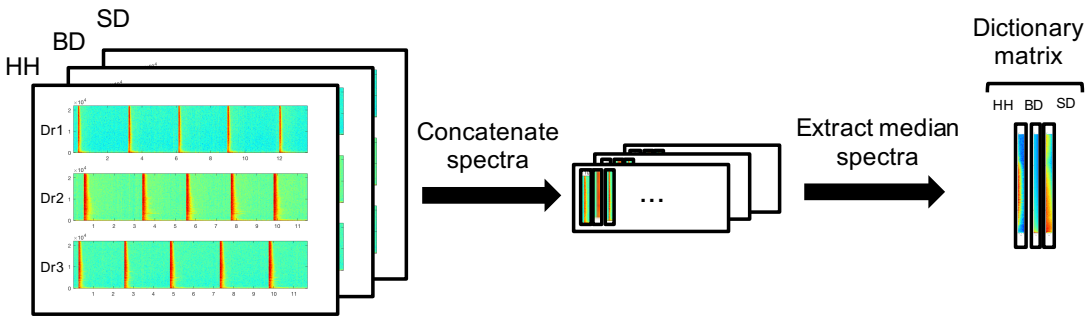


Figure 4.5: The process of building the predefined drum dictionary. Dr1, Dr2, and Dr3 are drummer 1, drummer 2, and drummer 3, respectively

drum class, one track per drummer is collected as training data. The onset position of these single hits was determined using the annotated ground truth. The template spectrum is a median spectrum of all individual events of one drum class in the training set. The templates are extracted for the three classes: HH, BD, and SD. The above mentioned process is visualized in Fig. 4.5.

To find the local maxima of the activation functions, a median filter as defined in Eq. 4.13 is used for peak picking. The filter length and the offset coefficient $\lambda$ of the median adaptive threshold are set to be 0.1 s and 0.12 for every track. The Matlab implementation of the presented system is available online.[1]

## 4.3 Evaluation

### 4.3.1 Data Preparation

The experiments have been conducted on the two major ADT datasets as introduced in Sect. 3.1. The first one is the *minus one* subset from the ENST drum dataset [94]. The minus one subset has 64 tracks of drum recordings with the corresponding accompaniments. Each track in this subset has an average duration of 55 s with varying style. The accompaniments are mixed with their corresponding drum tracks using a scaling factor of 1/3 and 2/3; this setting is consistent with several previous studies [19, 58].

[1] https://github.com/cwu307/NmfDrumToolbox

The second dataset, used for cross-dataset validation, is IDMT-SMT-Drums [74]. As mentioned in Sect. 3.1, this dataset provides isolated drum sounds for training. However, in the following experiments, the isolated sounds are not used. Note that MDB-Drums dataset described in Chapter 3 was not yet available at the time of the experiments, therefore, it was not included in the evaluation.

### 4.3.2 Evaluation Setup

The implemented system is evaluated for both DTD (drum only) and DTM (polyphonic mixtures). The same set of audio tracks is used with and without accompaniments. A three-fold cross-validation is applied to the evaluation process. Single drum hits collected from two drummers are used to train the system, and complete mixtures from the third drummer are used to test the system. The process repeats three times to test every drummer in the dataset. This process is the same as described in Paulus's study [58], and the purpose is to prevent the system from seeing the test data. Note that the training data used in the system are single drum hits, and the number of onsets is significantly fewer than the test data. The training data only consists of 10 to 12 single hits for each drum class. This is similar to the real-world use case, where the users may have access only to a limited number of training samples.

The evaluation metrics follow the standard calculation of the precision (P), recall (R), and F-measure (F), same as previously introduced in Sect. 2.3.6.

### 4.3.3 Evaluation Results

*Rank Independence*

In an initial test to determine the rank $r_{\mathrm{H}}$ of the PFNMF, $r_{\mathrm{H}} = 5, 10, 20, 40, 80, 160$ have been tested in polyphonic signals with and without a weighting matrix. As shown in Fig. 4.6, a general trend of decreasing performance can be observed when $r_{\mathrm{H}} > 5$ without a weighting matrix. With a weighting matrix, however, the performance slightly increases for both HH

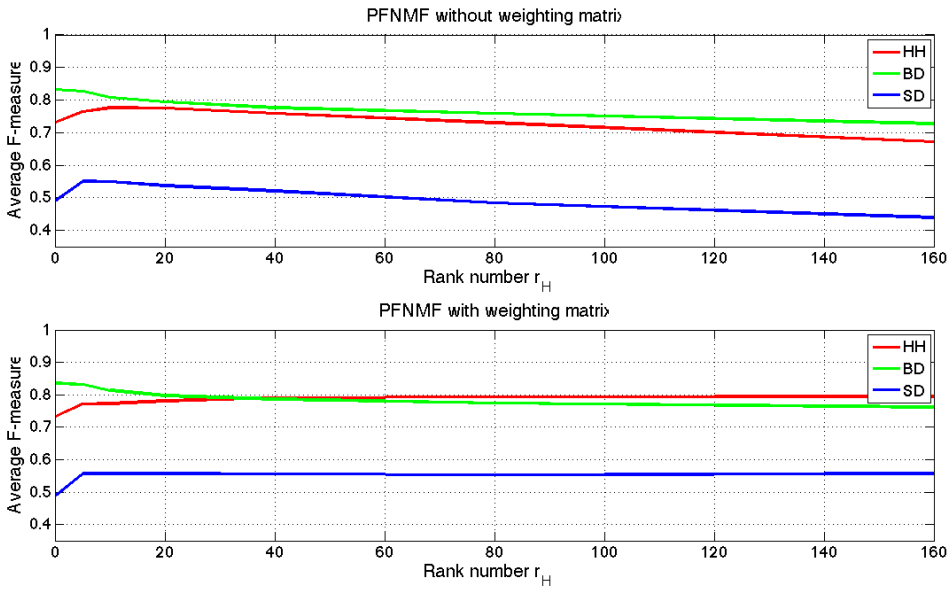


Figure 4.6: Average F-measure versus harmonic rank $r_{\mathrm{H}}$ in (Top) without weighting matrix (Bottom) with weighting matrix

and SD, and slightly decreases for BD as the $r_{\mathrm{H}}$ increases. The results demonstrate the robustness of the proposed system against the rank selection when a weighting matrix is introduced.

*Threshold Selection*

The transcription results can be obtained after applying onset detection on each drum activation (see Sect. 4.2.4). However, the performance varies according to the selection of the signal-adaptive threshold. To evaluate the influence of different thresholds, the average F-measure of all drums with different offset coefficient $\lambda$ on IDMT-SMT-Drums dataset is shown in Fig. 4.7. The results approximately follow a parabolic curve. This is in agreement with the findings of Dittmar et al. [74]. One major difference is that in most regions of the curve, both AM1 and AM2 outperform PFNMF. This verifies the template adaptation process does help the algorithm in the case of the unknown sounds (templates and the test signals are from two different datasets). The overall performance is slightly lower than the results reported by Dittmar et al. [74] due to the mismatch in templates and target signals.

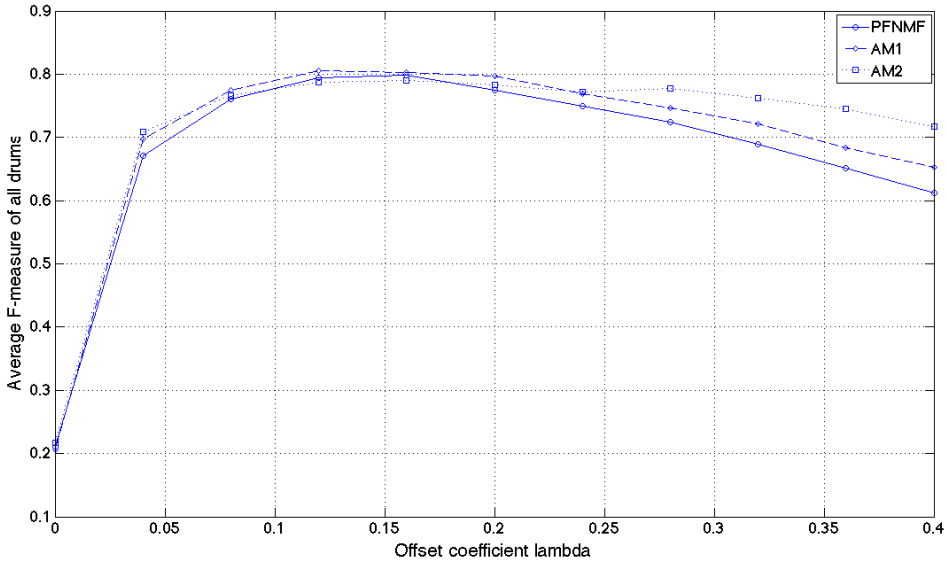


Figure 4.7: Evaluation results for IDMT-SMT-Drums dataset using (a) PFNMF (Solid circle) (b) AM1 (Dash diamond)(c) AM2 (Dotted square)

However, the F-measures of AM1 can reach $74.0\%$, $93.2\%$ and $73.4\%$ for HH, BD, SD, respectively, which indicates the applicability of the proposed method across datasets.

*Transcription Performance*

Table 4.1 shows the evaluation results on the sENST drum dataset *minus one* subset *without accompaniments*. For comparison, the results of Gillet et al. [19] and Paulus et al. [58] are also included. Both of the compared methods are data-driven. All the compared methods use the same dataset with identical mixing settings (1/3 for accompaniments and 2/3 for drum tracks). Since the target signals contain only drum sounds, the rank $r_{\mathrm{H}}$ can be small. In this experiment, $r_{\mathrm{H}}$ is set to 10 for absorbing drum sounds other than HH, BD and SD. The results show that PFNMF is able to transcribe drum events with an average F-measure of $77.9\%$ using AM2. This result is higher than the $73.8\%$ reported in [19], and at the same level as reported in [58].

Table 4.2 shows the evaluation results on ENST drum dataset *minus one* subset *with accompaniments*. The compared methods are the same as described above. Since the target

Table 4.1: Evaluation results for ENST drum dataset *minus one* subset **without** accompaniments. The best F-measure of each column is highlighted in bold.

| Method | Metric | HH | BD | SD | Mean |
|---|---|---|---|---|---|
| PFNMF | P | 0.918 | 0.886 | 0.825 | 0.876 |
| | R | 0.705 | 0.938 | 0.453 | 0.698 |
| | F | 0.797 | 0.911 | 0.585 | **0.764** |
| AM1 | P | 0.909 | 0.955 | 0.837 | 0.900 |
| | R | 0.682 | 0.927 | 0.473 | 0.694 |
| | F | 0.779 | **0.940** | 0.604 | **0.774** |
| AM2 | P | 0.928 | 0.914 | 0.854 | 0.898 |
| | R | 0.703 | 0.927 | 0.483 | 0.704 |
| | F | 0.799 | 0.920 | 0.617 | **0.779** |
| Gillet et al. [19] | P | 0.736 | 0.798 | 0.710 | 0.748 |
| | R | 0.865 | 0.700 | 0.642 | 0.735 |
| | F | 0.795 | 0.745 | **0.674** | **0.738** |
| Paulus et al. [58] | P | 0.838 | 0.941 | 0.750 | 0.806 |
| | R | 0.849 | 0.921 | 0.567 | 0.843 |
| | F | **0.843** | 0.930 | 0.645 | **0.779** |

signals contain both percussive and harmonic parts, $r_{\mathrm{H}}$ is set to 50. The results show that PFNMF achieves an average F-measure = $72.2\%$ using AM2, which is higher than $67.8\%$ [19] and at a similar range as the $72.7\%$, reported in [58].

### 4.4 Discussion

In general, PFNMF-based methods outperform [19] for all instruments except the SD. The possible reason is that many of the playing technique variations are applied to the snare (e.g., ghost note, rim shot, with/without snare on), and a single SD template cannot cover all the possibilities even with template adaptation. In the polyphonic dataset, PFNMF-based methods perform better on BD and SD but slightly worse on HH compared to the HMM based method [58]. Since Paulus et al. [58] only used the ENST dataset for both training and testing, there was a tendency of overfitting: in the ENST *minus one* subset, all three drummers were asked to play with the same set of accompaniments. Even with the different drum tracks, the resulting mixtures could still be similar. A cross-dateset validation would be necessary to verify the generality of the system.

Table 4.2: Evaluation results for ENST drum dataset *minus one* subset **with** accompaniments. The best F-measure of each column is highlighted in bold.

| Method | Metric | HH | BD | SD | Mean |
|---|---|---|---|---|---|
| PFNMF | P | 0.902 | 0.714 | 0.684 | 0.766 |
| | R | 0.706 | 0.862 | 0.464 | 0.677 |
| | F | 0.792 | 0.781 | 0.552 | **0.708** |
| AM1 | P | 0.904 | 0.781 | 0.758 | 0.814 |
| | R | 0.679 | 0.856 | 0.45 | 0.661 |
| | F | 0.775 | **0.816** | 0.564 | **0.719** |
| AM2 | P | 0.908 | 0.774 | 0.726 | 0.802 |
| | R | 0.694 | 0.855 | 0.466 | 0.671 |
| | F | 0.786 | 0.812 | 0.567 | **0.722** |
| Gillet et al. [19] | P | 0.702 | 0.744 | 0.619 | 0.688 |
| | R | 0.818 | 0.653 | 0.552 | 0.674 |
| | F | 0.755 | 0.695 | **0.583** | **0.678** |
| Paulus et al. [58] | P | 0.847 | 0.802 | 0.663 | 0.770 |
| | R | 0.826 | 0.815 | 0.453 | 0.698 |
| | F | **0.836** | 0.808 | 0.538 | **0.727** |

For all the methods, the performances drop from the monophonic to the polyphonic dataset, especially for BD and SD. This is an unsurprising trend. The less prominent decrease for HH might be due to the fact that the typical frequency range of HH is more separated from other instruments than BD and SD, thus is more robust against the presence of tonal sounds. In the case of template adaptation, a general trend of increase in precision and decrease in recall can be observed. One explanation is that once a better representation of the drum templates is found, the system might become more selective, leading toward a reduction in both false positives and true positives.

AM1 seems to perform better than AM2 on BD in both monophonic and polyphonic dataset. One possible explanation is that bass drum usually appears on the downbeats, which tends to have higher correlation with other entries in harmonic activation matrix. This means BD has a higher chance of being adapted to better templates using AM1. AM2 uses a more generalized adaptation process and performs better on HH and SD. However, it is more computationally demanding since it adapts the templates constantly, whereas AM1 only adapts when the correlation is above the threshold. To summarize, both template adaptation

methods perform at a similar level, and the best fit of either method for specific types of music still needs to be investigated.

## 4.5 Conclusion

In this chapter, a drum transcription system for both DTD and DTM using PFNMF with template adaptation is presented. The system is robust against rank changes, and the evaluation results show that the two presented template adaptation methods improve the precision of the system, leading towards better performance.

The presented system has the following advantages: First, the system only requires a few training samples for template extraction, and these templates can adapt toward the target sources gradually. This makes the system applicable to realistic use cases. Second, adjustment of the parameter $r_{\mathrm{H}}$ allows the algorithm to work with polyphonic music, and the use of a weighting matrix prevents the performance from dropping as $r_{\mathrm{H}}$ increases. Third, the cross-dataset evaluation results indicate a robustness against template mismatches, possibly allowing the application in situations with minimum prior knowledge.

PFNMF has been shown to perform in a comparable range with more complex models such as RNNs [79] with a good generalizabilty. However, since the algorithm assumes the signal to be the linear combinations of different sound sources, it might not be as flexible as models with non-linearity (e.g., SVM with RBF kernels or neural networks). Additionally, the template adaption only works if the initial templates provide a reasonable starting point. When the initial templates deviate too far from the actual spectra of the drum sounds in the target signal, the performance of the system might be adversely impacted and could not recover using the template adaption methods.

# CHAPTER 5
# DRUM TRANSCRIPTION WITH UNLABELED DATA

The content of this chapter has been published in the following publications:

- **Chih-Wei Wu and Alexander Lerch, "Automatic Drum Transcription using Student – Teacher Learning Paradigm with Unlabeled Music Data," Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), 2017.**

- **Chih-Wei Wu and Alexander Lerch, "From Labeled to Unlabeled Data – On the Data Challenge in Automatic Drum Transcription," Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), 2018.**

As opposed to designing an algorithm that requires less training data, this chapter explores a different strategy for overcoming the data insufficiency. Particularly, the usefulness of unlabeled music data is investigated in the context of ADT. This chapter connects to RQ3, which is to find the promising directions for ADT systems to benefit from unlabeled data. Since unlabeled data does not introduce the additional cost of manual annotations, it is an ideal resource for data-driven ADT systems to improve in a scalable way. To incorporate unlabeled data in general ADT systems, two learning paradigms that harness information from unlabeled data are integrated into two major types of ADT systems, and their effectiveness are evaluated in the DTM setting. More details are described in the following sections.

## 5.1 Introduction

The ADT system described in the previous chapter was designed to approach the DTM task with minimum prior knowledge required; it is generally desirable when the data availability is concerned. However, for the future advancement of ADT research, it is also important to consider a complex system that has more potential to grow with a larger amount of data. In this regard, data-driven systems are promising for leading towards a significant breakthrough when the data is sufficient. For example, with the contributions of large-scale datasets such as ImageNet [103] and CIFAR-10 [104], training a complex model such as CNNs [105] with good generalizability became possible and led to a paradigm shift in computer vision. However, the direct translation of these successful models to the field of MIR is difficult, mostly due to the absence of large-scale music datasets.

Different strategies have been proposed previously to address the data challenge in MIR tasks. Generally speaking, they could fit inito one of the following settings (i) *Supervised* and (ii) *Unsupervised* methods. This categorization was derived from the major paradigms in statistical machine learning [106], which can be briefly defined as follows: Let $x_i \in X$ for $i = \{1, ..., n\}$ be a set of $n$ samples and $y_i \in Y$ be their corresponding labels, the goal of the first paradigm is to find the best function $f(X) = Y$ that maps $X$ to $Y$ and solves the underlying tasks. When $y_i$ is a discrete label, the task is known as classification; when $y_i$ is a continuous value, it is known as regression. The process of finding the best function is known as training. Since the first paradigm has the access to the labels (ground truths), the training process is guided and thus called supervised learning. In the second paradigm, on the other hand, the algorithm does not have the access to the labels. With only $X$ available, the goal is to find underlying structure of the data without any guidance. Common forms of the unsupervised learning problems are clustering and dimensionality reduction.

### 5.1.1 Supervised Methods

In the supervised setting, techniques that build upon the existing labeled data have been proposed. For example, in *transfer learning* for MIR tasks [107], a CNN trained on a task with sufficient data can be used to derive features for another task with limited data. This method alleviates the data insufficiency by re-using the effective models in the similar domains. *Data augmentation*, a technique to increase diversity of training data through music-related deformations (e.g., time-stretching, pitch shifting, or distortion) and synthesis, has been successfully applied in MIR tasks [108] and in ADT specifically [11, 81]. However, these techniques still require a reasonably sized correctly annotated dataset as a starting point, which can be a challenge in certain scenarios.

### 5.1.2 Unsupervised Methods

In the unsupervised setting, the direction for addressing the data scarcity is to use unlabeled data. Intuitively, a large collection of unlabeled data can be helpful in deriving more generalized features. This is the main concept of unsupervised *feature learning*, and it can be implemented with algorithms such as Sparse Coding [109], Deep Belief Networks (DBNs) [110], and Auto-encoders [111]. For example, Raina et al. [109] proposed a method called self-taught learning that aims to learn better feature representations from the unlabeled data using Sparse Coding [112]. Once the features are learned from the unlabeled data, the same set of features can be extracted from the labeled data for training a standard classifier such as Support Vector Machine (SVM). Experimental results showed that this method is effective for improving the performance of varioius tasks including a music genre classification system [109]. Similarly, Jao et al. [113] proposed to learn sparse codes from the unlabeled data as feature representations and achieved improvement on the music auto tagging task. The same concept can also be extended to other feature learning techniques such as NMF [114]. In [110], Hamel and Eck proposed to learn features from music audio with DBNs and used these features for a music genre classification task. The results showed

that these learned features were able to achieve comparable performance with the commonly used audio features, which are manually designed based on domain knowledge (see [22] for an introduction to these features).

In general, the idea of unsupervised feature learning is appealing, and it has been proven successful in different fields such as image and video processing [115, 111, 116]. However, since the process is unsupervised, it is difficult to determine whether the feature learner is able to capture the essential information. In some cases, the learned features are hard to interpret, and their effectiveness can only be determined based on the performance on a specific task. This indirect assessment puts the generality of the model at risk. Additionally, the ideal input representation for feature learning is another open parameter that is difficult to configure. In one study [117], it is observed that raw input representation such as STFT is suitable for feature learning; in another study [118], it is found that an input representation that incorporates minimum domain knowledge can be a better choice than STFT. In the end, the optimal setting for feature learning is still task-dependent and requires heuristics.

### 5.1.3 Semi-supervised Methods

When both the labeled and unlabeled data are available, a third type of learning problem, known as *Semi-supervised Learning* [119], can be formulated. Let $X = \{X_u, X_l\}$ and $Y = \{Y_l\}$ in which $X_u$ is the unlabeled data, $X_l$ is the labeled data, and $Y_l$ is the corresponding label of $X_l$, the goal of semi-supervised learning is to learn the best function that maps $X$ to $Y$, which outperforms the supervised setting when only $X_l$ and $Y_l$ are available. Generally speaking, semi-supervised learning is useful when $X_u$ carries relevant information that will help the inference of $Y$ given $X$, and it is most effective when the number of $X_u$ is considerably larger than $X_l$. This assumption is usually valid in the context of music data, for the labeled data is rare compared to the unlabeled data. Therefore, semi-supervised learning has a great potential to improve the performances of the existing MIR systems. For example, Wu et al. proposed to apply semi-supervised learning in the task of emotion

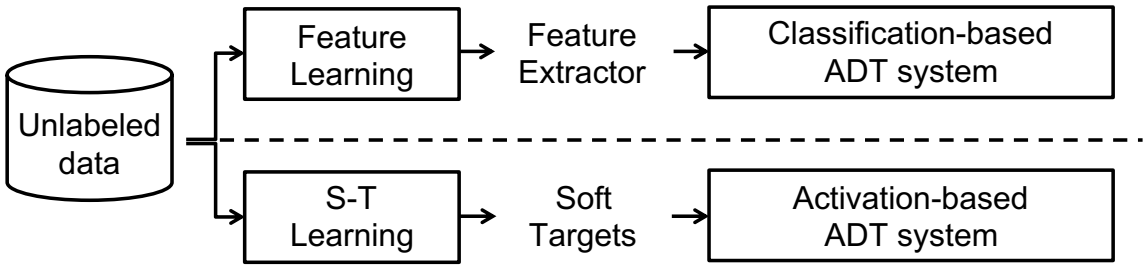


Figure 5.1: The overview of the evaluated paradigms for integrating unlabeled data to two major ADT approaches

recognition with social tagging [120]. Using only a small amount of labeled data, the system was able to achieve comparable results with the supervised approaches.

More recently, the *student-teacher learning* paradigm has also emerged as an interesting concept to incorporate unlabeled data. Referred to by Hinton et al. as "knowledge distillation" [121], this paradigm transfers the knowledge of a teacher model to a student model using the soft-targets generated by the teacher. Typically, a teacher model is an expert system that is pre-trained to perform a specific task; such a model can sometimes be slow and cumbersome due to the model complexity. A student model, on the other hand, is a lightweight system with lower complexity that aims to mimic the teacher. As opposed to learning from the hard targets (i.e., ground truth), the student learns from the "dark knowledge" residing in the soft-targets, which can be created using either labeled or unlabeled data [122]. A successful student model can reduce the complexity of the original teacher model without significant performance loss. The main purpose of this paradigm is to compress large models such as DNN into a smaller one and run on the devices with low computational power. However, some studies even report superior performance of the student models [123, 124, 125]. As a result, this paradigm provides an interesting way of improving the existing systems.

## 5.2 Method

### 5.2.1 Overview

To connect general ADT systems to the abundant resources of unlabeled data, this chapter investigates the application of *feature learning* and *student-teacher learning* to *Classification-based* and *Activation-based* ADT systems, respectively. As summarized in Table 2.1, the majority of the existing ADT systems until now fall into the categories of *Classification-based* and *Activation-based* ADT systems. To demonstrate the viability of improving general ADT systems with unlabeled data, we consider both types of systems in our experiments.

Fig. 5.1 shows the two paradigms for integrating unlabeled data to ADT systems as investigated in this chapter. The feature learning paradigm is designed for *Classification-based* ADT systems. In this paradigm, the unlabeled data is used to derive a feature extractor using an unsupervised feature learning algorithm. The resulting feature extractor is then integrated into a generic *Classification-based* ADT framework. The student-teacher learning paradigm is suitable for *Activation-based* ADT systems. This paradigm uses teacher models and unlabeled data to generate soft-targets; these soft-targets play the important role of connecting any *Activation-based* system with unlabeled data and enable the training of the student model. In the following sections, more details of both paradigms are presented.

### 5.2.2 Feature Learning

The flowchart in Fig. 5.2 shows the feature learning paradigm for ADT, including both training and testing. The training phase starts with the training of a feature extractor using the unlabeled data. In this case, the encoder part of a Convolutional Auto-encoder (CAE) is used as the feature extractor. A generic *Classification-based* ADT system is then constructed with the following steps: first, the features are extracted from the audio signals using the pre-trained feature extractor. Second, the onset locations are determined by using the ground truth annotations while training. Finally, the feature vectors around the onset locations are

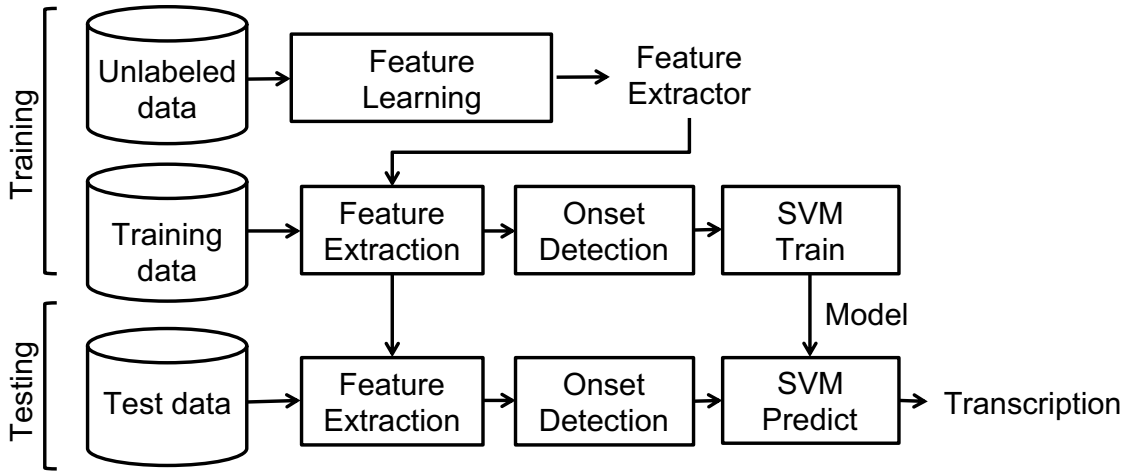


Figure 5.2: The flowchart of the feature learning paradigm for ADT

collected and used to train three binary classifiers for HH, BD, and SD, respectively. The classifiers are Support Vector Machines (SVMs). In the testing phase, the same pipeline is used except for the onset detection step, which uses an onset detector instead of the ground truth locations. Finally, the presence of each drum can be predicted using the pre-trained SVMs.

*Convolutional Auto-encoder*

The architecture of the implemented CAE is shown in Fig. 5.3. The design of this architecture is based on the work of Choi et al. [107]. In Choi's work, the CNN model is trained in a supervised fashion for an audio auto-tagging task; after training, the CNN can be used as a feature extractor and is shown to be effective for several audio related tasks. In this work, a similar CNN model is adopted with minor adaptations. The modifications include: (i) a symmetric architecture that has the same input and output dimensionality and (ii) a bottleneck structure that enforces the concentration of the essential information at the bottleneck layer. The input $X$ of the CAE is a Mel-spectrogram, and the output $X'$ is the reconstruction of $X$. The encoder consists of four convolutional layers with $\{32, 16, 8, 4\}$ channels of $3 \times 3$ kernels, accordingly. Each convolutional layer is followed by a batch normalization [126] layer and a max-pooling layer of (2, 1). This design maintains the

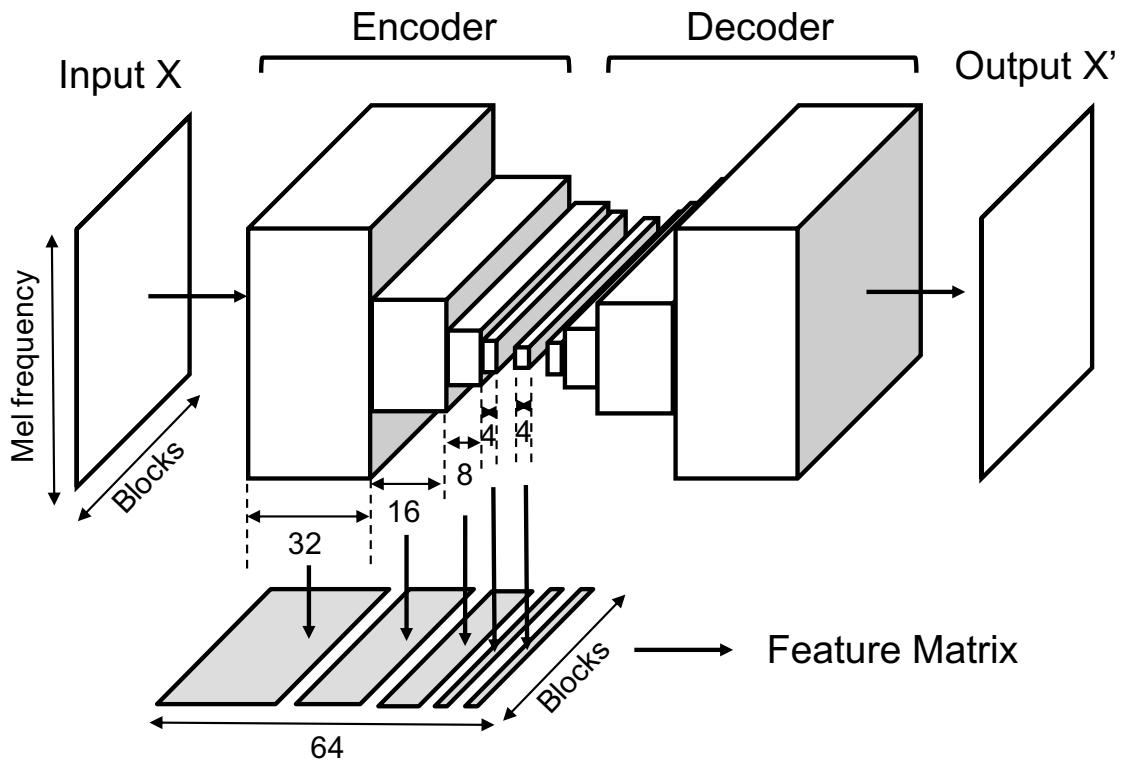


Figure 5.3: The architecture of the proposed CAE for unsupervised feature learning. The input $X$ is a $128 \times N$ Mel-spectrogram.

temporal resolution, allowing the extraction of block-wise features. The bottleneck layer is also a convolutional layer with 4 channels of $3 \times 3$ kernels. All non-linear units are Rectified Linear Units (ReLUs). The structure of the decoder is symmetric to the encoder with the max-pooling layers replaced by the up-sampling layers. The CAE is trained to minimize the Mean Squared Error (MSE) between $X$ and $X'$ using a gradient-descent-based optimization process, and the number of training epochs is $30$.

The feature extraction process, as shown in Fig. 5.3, is inspired by the method proposed by Choi et al. [107]: first, the intermediate outputs from all the layers in the encoder (including the bottleneck layer) are computed. Next, these outputs are aggregated across the Mel-frequency axis through averaging. Finally, the aggregated outputs are stacked into a $64 \times N$ feature matrix, where $N$ is the number of blocks. To derive the final feature vector at each block, the feature vectors from the current block and the following two blocks are

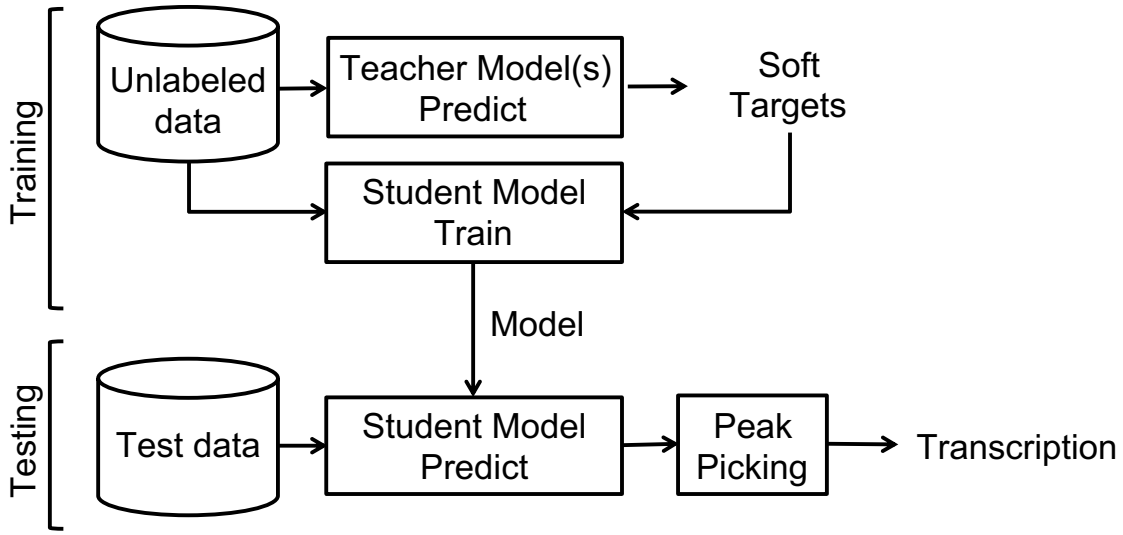


Figure 5.4: The flowchart of the student-teacher learning paradigm for ADT

spliced together to capture the temporal variations of the event. This leads to a final feature vector with a dimensionality $d = 3 \times F$, in which $F$ is the number of features (i.e., 64).

In addition to the learned features, a set of baseline features consisting of 20 Mel Frequency Cepstral Coefficients (MFCCs) and their first and second derivatives is also included. As a result, the baseline feature vector has a dimensionality $d = 3 \times 60 = 180$ after the feature splicing.

### 5.2.3 Student Teacher Learning

Figure 5.4 shows the flowchart of the student-teacher learning paradigm for ADT. In the training phase, the teacher models are used to analyze the unlabeled data and generate the soft-targets. These soft-targets, used as pseudo ground truth to train a student model, contain the activation functions for the different drums. When multiple teachers are presented, the student model can be trained by iteratively passing the unlabeled data and its corresponding soft-targets from each teacher. The student model is trained by minimizing the MSE between the soft-targets and the model outputs. In the testing phase, the trained student model processes the test data and generates the corresponding activation functions. The estimated locations of drum hits are identified with a simple peak picking process. More

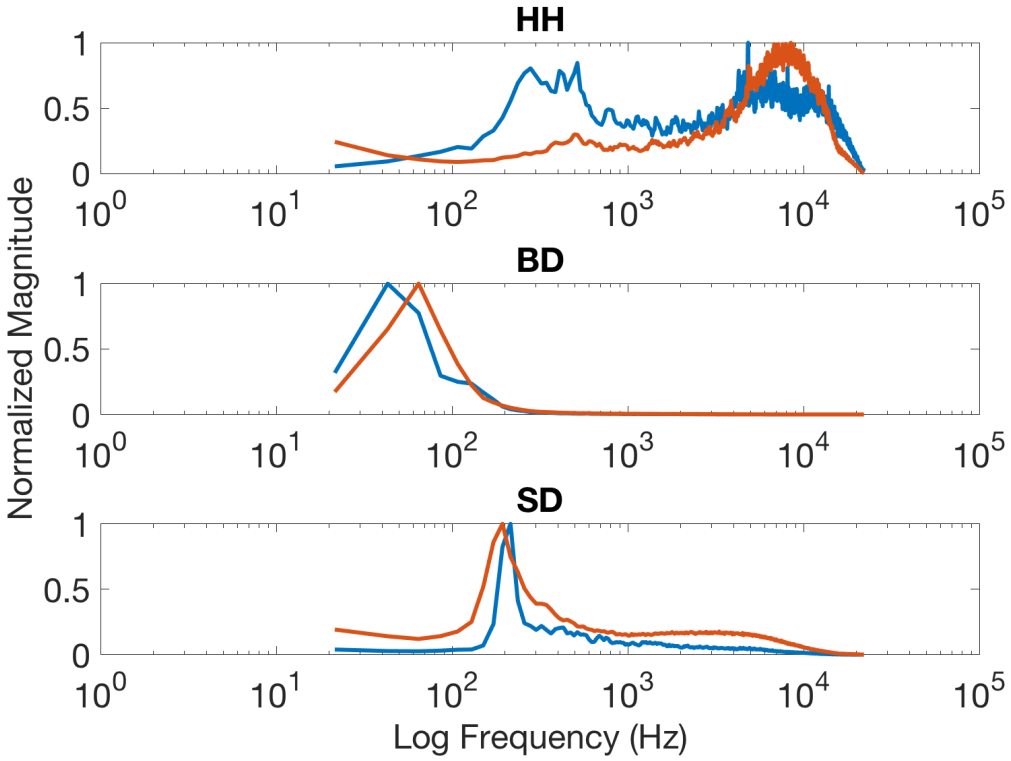


Figure 5.5: Comparison of the drum templates extracted from the IDMT-SMT-Drums (blue line) and 200 Drum Machines (red line) dataset for different instruments

elaborate descriptions of the teacher and student models can be found in the following sections.

*Teacher Model*

The teacher model is the PFNMF system presented in Chapter 4. This NMF-based ADT system is chosen for its simplicity, its lack of need for substantial amounts of training data, as well as the adaptability to polyphonic mixtures; it extends the basic NMF model to PFNMF by assuming the co-existence of both percussive and harmonic components in the audio signals. As described in Sect. 4.2.2, once the signal is decomposed, the activation function $H_{\mathrm{D}}(r,:)$ of each individual drum can be extracted, in which $r = \{1, 2, 3\}$ is the instrument index that corresponds to HH, BD, and SD, respectively. These activation functions can be interpreted as the activity level of each instrument over time, and a sharp peak indicates the

presence of a single drum hit.

The conversion of the resulting activation functions into the soft targets takes another step of standard min-max scaling across the training data for each instrument; this process scales the soft targets to a numerical range between 0 and 1 and ensures the compatibility between the soft targets and the student model output. Finally, to introduce diversity into the soft targets, two PFNMF systems are created by initializing the algorithm with two different sets of drum dictionaries, forming an ensemble-like scenario that could potentially lead to better student performance. The extraction of these two drum templates takes place on two publicly available drum datasets (see Sect. 3.1), namely the IDMT-SMT-Drums dataset [74] and 200 Drum Machines [92]. As shown in Fig. 5.5, these two sets of templates exhibit capabilities of representing different types of drum sounds, thus adding diversity to this learning paradigm.

The construction of the drum dictionary is the same as described in Sect. 4.2.5. It should be noted that none of these two datasets are used during the testing in order to ensure the generality of the proposed approach.

*Student Model*

The proposed student model is a fully connected, feed-forward DNN with three hidden layers. A neural network is a graphical model that comprises multiple layers of interconnected non-linear units (i.e., neurons). The basic formulation of a neuron can be expressed in Eq. (5.1)

$$a_k^l = g\left(\sum_{j=1}^{M} W_j a_j^{l-1} + b_j^{l-1}\right), \tag{5.1}$$

in which $a$ is the activation of the neuron, $W$ is the weight matrix, $b$ is the bias matrix, $l$ is the layer index, $j$ is the index of input neuron, and $k$ is the index of output neuron; $g()$ is usually a non-linear function such as a *sigmoid*, *tanh* or *ReLU*. When multiple layers of neurons are stacked, the model creates a non-linear transformation from the input to the output, which allows the model to approximate any arbitrary function with great flexibility.

The architecture of the DNN is as follows: the input layer contains $1025$ neurons that correspond to the size of the input representation. The first hidden layer comprises of $1025$ neurons of *tanh* units with batch normalization. The second and third hidden layers have $512$ and $32$ neurons with ReLU units, respectively. Finally, the output layer consists of 3 neurons with *sigmoid* units that represent the activities of three different drums (i.e., HH, SD, and BD). The architecture and type of neurons are selected based on the results of smaller-scale preliminary experiments, and the fully connected layers are chosen for their simplicity and generality. To solve the optimization problem of learning the weights $W$ in a DNN, a stochastic gradient descent based optimization method, Adam [127], is selected as the optimizer. The student neural network is configured as a regressor that minimizes the MSE between its output and the soft targets. A mini-batch consisting of $640$ instances is used for training, and the early stopping technique is applied to stop the training process when the loss decrease is less than $10^{-6}$ for three consecutive epochs.

### 5.2.4 Implementation

The main input representations for both paradigms are derived from the magnitude spectrogram of the Short Time Fourier Transform (STFT), which is computed using a block size of 2048 and a hop size of 512 samples with Hann window. All of the audio signals are normalized to a range between 1 and -1, down-mixed to mono, and resampled to 44.1 kHz prior to the computation of STFT.

For the feature learning paradigm, both the Mel-spectrogram in dB scale with 128 bins and the MFCCs are computed using librosa [128], a Python library for audio signal processing. The onset detection is implemented using the *CNNOnsetProcessor* from Madmom [97]. Additionally, the implementation of Linear SVMs from scikit-learn [129], a Python library for machine learning, is used. A grid search on the penalty parameter $C$ within $\{0.1, 1, 10, 100, 1000\}$ is performed to optimize the performance of the SVMs.

For the student-teacher learning paradigm, the teacher models are implemented using

the PFNMF function from NmfDrumToolbox.[1] The peak-picking methods are described in Sect. 4.2.4, and the parameters for the adaptive threshold are the same as in Sect. 4.2.5.

The neural networks in both paradigms are implemented using Keras[2] and the Tensorflow [130] backend. The weights are randomly initialized with normal distributions, and the parameters of the optimizers are set to default. The source code used in this chapter is available on Github.[3]

## 5.3 Evaluation

The evaluation of the proposed methods require both labeled and unlabeled data. In the following sections, the preparation of these datasets are presented. Also, the experiment setup, including the evaluated systems and the experiment configurations, is introduced in more detail.

### 5.3.1 Unlabeled Dataset

The collection of the unlabeled data is a crucial step for ensuring a successful learning process. Generally speaking, the unlabeled dataset should have following attributes:

(i) the collection should contain drums whenever possible,

(ii) the collection should be diverse in terms of music genres or playing styles,

(iii) the collection should contain no duplicates, and

(iv) the collection should be as consistent as possible in terms of audio quality.

To build a collection that meets the above-mentioned criteria, a software tool is built, allowing the compilation of a list of songs from the Billboard Chart[4] and the retrieval of these songs from Youtube. This dataset consists of six musical genres, including R&B/HipHop,

[1] https://github.com/cwu307/NmfDrumToolbox, last access 2018/03/27
[2] https://keras.io, last access 2018/03/27
[3] https://github.com/cwu307/ADT_with_unlabeledData, last access 2018/04/15
[4] https://www.billboard.com/charts, last access 2018/03/27

Pop, Rock, Latin, Alternative, and Dance/Electronic. Each genre has 1900 songs, which leads to a collection of 11400 songs. All the songs are cross-checked for duplicates and converted to mp3 format with a sampling rate of 44.1 kHz. In our experiments, this dataset is further split into training, validation, and testing set with a percentage of 70%, 15%, and 15%, respectively. To speed up the process while maintaining the diversity, only a 30 s segment is extracted from each song for training. The segment starts in the middle of the song to avoid potential inactivity at the beginning. As a result, the entire training set has a total duration of 66.5 hrs, which is significantly larger than any existing ADT dataset. The list of songs and links are available on Github.[5]

### 5.3.2 Labeled Dataset

To evaluate the methods described in this chapter, four different labeled datasets for DTM are used: (i) the popular ENST-Drums (referred to as ENST) [94], (ii) the MIREX 2005 (referred to as m2005)(iii) the MDB-Drums (referred to as MDB) [17], and (iv) the RBMA dataset [83]. The latter three public sets have been used in the 2017 Music Information Retrieval Evaluation eXchange (MIREX)[6] drum transcription task. Note that m2005 and RBMA are currently only available to the MIREX participants.

More details on these datasets are presented as follows:

(i) *ENST minus one* subset is the same dataset used in Chapter 4 (see Sect. 4.3.1). This subset is the most popular dataset for the DTM task and has been considered as the benchmark dataset. The total duration of this subset is approximately 1 hr.

(ii) *m2005* was originally collected for the first MIREX drum transcription task back in 2005 and re-released for MIREX 2017. The public set includes 23 recordings contributed from all the participants of MIREX 2005. While covering a variety of musical genres, Japanese-pop has the highest presence in this dataset with 10

[5] https://github.com/cwu307/unlabeledDrumDataset, last access: 2018/04/15
[6] http://www.music-ir.org/mirex/wiki/2017, last access 2018/03/27

recordings. The average duration of this dataset is 125 s, and the total duration is approximately 0.8 hr.

(iii) *MDB* consists of 23 recordings of the MusicDelta subset from the MEDLEYDB dataset [96]. These recordings include a variety of musical genres such as Rock, Country, Disco, Reggae, and Jazz. The average duration of the recordings is 54 s, and the total duration of the dataset is roughly 0.35 hr. Similar to *ENST*, this dataset contains multi-track files as well as the full mixtures. For the following experiments, the full-mixtures are directly used without any adjustment of the mixing levels.

(iv) *RBMA* was released as part of the public set for the MIREX 2017 drum transcription task. This public set includes 27 recordings featuring mostly Electronic Dance Music (EDM). The average duration of the tracks is 230 s, and the total duration of the dataset is about 1.7 hr. Since this dataset focuses on electronic music, it contains electronic drum sounds that are distinctively different from the other three datasets.

In total, there are 137 files with annotations available for evaluation. All files have a sampling rate of 44.1 kHz.

### 5.3.3 Evaluation Setup

This section includes the evaluation of 9 ADT systems, comprising 4 systems for the feature learning paradigm and 5 systems for the student-teacher learning paradigm. The evaluation metrics in the experiments are Precision (P), Recall (R), and F-measure (F) as introduced in Sect. 2.3.6. Only the averaged F-measure (either across different datasets or systems) is reported for better clarity. These metrics are implemented using *mir_eval*, a Python library of common MIR metrics [131]. The configuration of these systems is described as follows:

For the feature learning paradigm, the 4 systems are differentiated by their features. These features are:

(i) MFCC: this set of features has shown its effectiveness in previous ADT studies (see Table 2.1). Therefore, it is included as a baseline.

(ii) CONV-RANDOM: this set of features is extracted using the proposed CAE architecture with all the weights randomly initialized without further training. This is another baseline inspired by [107] to serve as a sanity check for the effectiveness of the unsupervised training process.

(iii) CONV-AE: this is the set of features extracted from the proposed CAE after training. During the training procedure, the original input is used as the target for optimization. In other words, the CAE is trained to reconstruct the input.

(iv) CONV-DAE: this set of features is similar to CONV-AE except for the optimization target. In this case, a processed input is used as the target. Specifically, the percussive component from the Harmonic Percussive Source Separation (HPSS) [24] algorithm is used, and the CAE is trained to approximate the percussive component. This configuration is inspired by the concept of the Denoising Autoencoder (DAE) [132] and is designed to encourage the extraction of drum-related features.

The teacher models for student-teacher learning paradigm are described in Sect. 5.2.3. The 3 student models can be differentiated by their training data. The systems are:

(i) PFNMF (SMT): a teacher PFNMF initialized with the drum templates extracted from the IDMT-SMT-Drums dataset [74].

(ii) PFNMF (200D): a teacher PFNMF initialized with the drum templates extracted from the 200 Drum Machine dataset [92]

(iii) FC-200: a fully-connected student DNN trained with a subset of the unlabeled dataset, which consists of 200 randomly selected songs from each genre.

(iv) FC-ALL: a fully-connected student DNN trained with all the songs from all genres.

(v) FC-ALL (ALT): a fully connected student DNN trained with all the songs from only the "Alternative" genre. This particular genre is selected for its superior performance

in preliminary tests.

Based on these 9 systems, the following experiments are conducted:

**E1: Experiment 1** aims to examine the variance of the labeled datasets. For each dataset, the averaged F-measures across all 9 systems are reported.

**E2: Experiment 2** aims to evaluate the usefulness of unlabeled data for *Classification-based* ADT systems using the feature learning paradigm. For each system, the averaged F-measures across all the datasets are reported. Note that for feature learning paradigm, a cross-dataset validation process is performed (e.g., train on three datasets and test on the remaining one) in order to train the binary classifiers (see Sect. 5.2.2 for more details).

**E3: Experiment 3** aims to evaluate the usefulness of unlabeled data for *Activation-based* ADT systems using the student-teacher learning paradigm. For each system, the averaged F-measures across all the datasets are reported. Since the student model does not need additional labeled data for training, the cross-dataset validation is unnecessary.

### 5.3.4 Evaluation Results

Figure 5.6 shows the evaluation result of **E1**. On average, all systems tend to perform the best on *ENST* and the worst on *RBMA*. For some instruments, this gap can be as large as 22% in F-measure. There are two possible reasons for the good performance on *ENST*. First, as many ADT systems, including *Classification-based* and *Activation-based*, have been developed and evaluated on *ENST*, there could be potential bias towards this dataset. Second, the *ENST* dataset might be relatively simple compared with the others. A closer examination of the dataset shows the lack of singing voices and the dominance of MIDI synthesized accompaniments, which could potentially over-simplify the ADT problem. The relative poor performance on the *RBMA* dataset might be related to its focus on EDM; the

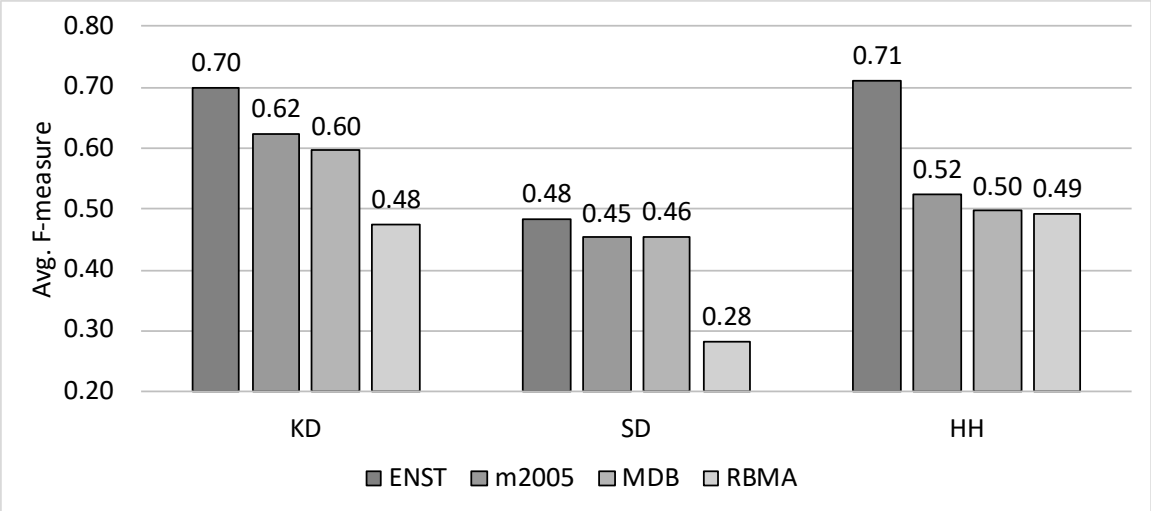


Figure 5.6: The evaluation results of all labeled datasets with averaged F-measure across all systems.

Table 5.1: Evaluation results of the feature-learning-paradigm-based systems.

| **Experiments** | | **Averaged F-measure** | | |
|---|---|---|---|---|
| **Role** | **System** | **HH** | **BD** | **SD** |
| Baseline | MFCC | 0.61 | **0.62** | 0.40 |
| Baseline | CONV-RANDOM | 0.61 | 0.54 | 0.39 |
| Evaluated | CONV-AE | 0.61 | **0.62** | **0.42** |
| Evaluated | CONV-DAE | 0.61 | 0.61 | **0.42** |

electronic drum sounds with strong audio effects could possibly increase the difficulty for ADT. This seems to be especially true in case of the SD. Overall, the results show that the evaluated systems leave much room for optimization; since many of the parameters in these systems are not extensively tuned, this result is to be expected. However, this also reflects the challenge of building an ADT system that is easily generalizable.

The results of **E2** are shown in Table 5.1. The following trends can be observed: first, the unlabeled data seems to be helpful in *Classification-based* ADT systems. A direct comparison between CONV-AE and MFCC shows that the features learned from unlabeled data seem to slightly improve for SD while achieving equal performance on HH and BD. Second, the unsupervised training process is useful for deriving better features. Compared to CONV-RANDOM, both CONV-AE and CONV-DAE show improvements on nearly

Table 5.2: Evaluation results of the student-teacher-paradigm-based systems.

| **Experiments** | | **Averaged F-measure** | | |
|---|---|---|---|---|
| **Role** | **System** | **HH** | **BD** | **SD** |
| Teacher | PFNMF (SMT) | 0.47 | 0.61 | **0.45** |
| Teacher | PFNMF (200D) | 0.47 | **0.67** | 0.40 |
| Student | FC-200 | **0.56** | 0.57 | 0.44 |
| Student | FC-ALL | 0.53 | 0.59 | 0.42 |
| Student | FC-ALL (ALT) | 0.55 | 0.58 | 0.44 |

all instruments, indicating the advantage of the training process. Third, the DAE-inspired training process does not lead to improvements for ADT. This is shown by the almost equivalent results from CONV-AE and CONV-DAE. Since HPSS also introduces artifacts, it might not be the most ideal method for this task; experimentation with other source separation algorithms might provide more insights. All of the systems achieve similar performance on HH. One possible reason could be its distinctive sound characteristics. HH usually features a frequency range that is easily separable from the other instruments; this information might be relatively simple to capture even with different feature representations. As a result, different sets of features perform similarly on this instrument.

Table 5.2 shows the results of **E3**. The general trends can be summarized as follows: first, the student-teacher learning seems to be useful for *Activation-based* ADT systems as all students show a noticeable improvement on HH over the teacher models. This observation consolidates the preliminary finding reported in [125]. Second, more unlabeled data do not necessarily lead to better results. For example, FC-200 and FC-ALL (ALT) both outperform FC-ALL on HH and SD. Since the student model is a simple feed-forward DNN, the lack of model capacity could limit its potential for further improvement as the data size grows. Experiments using other student models (e.g., CNNs and RNNs) would be necessary for confirmation. Third, the student models seem to struggle on BD. A detailed examination on the individual results from each dataset shows that teachers and students are mostly comparable on BD except for *RBMA*. This is possibly due to the challenging nature of *RBMA* as discussed in **E1**. However, further investigation might be needed before drawing

Table 5.3: Significance check of the most improved pair from each paradigm.

| **Paradigm** | | **Feature Learning** | **Student-Teacher Learning** |
|---|---|---|---|
| **Compared Systems** | | CONV-AE vs. MFCC | FC-200 vs. PFNMF (SMT) |
| **Instrument** | | SD | HH |
| **Improved** | **# Files** | 70/137 | 78/137 |
| | **Avg. Gain (%)** | 6.5 | 13.8 |
| **Deteriorated** | **# Files** | 40/137 | 44/137 |
| | **Avg. Loss (%)** | -4.6 | -7.6 |

conclusions.

## 5.4 Discussion

The results of **E2** and **E3** show that feature learning and student-teacher learning paradigms are able to improve the performance on SD and HH, respectively. In light of these results, an interesting question is: "Are these improvements significant?" In an attempt to answer this question, two pairs of systems are selected for further analysis. Each pair consists of the best baseline and the best evaluated system of each paradigm. A t-test is performed on each pair by comparing their results on all 137 files. Both pairs show significant differences with $p \ll 0.05$ for both t-tests. Furthermore, the number of improved and deteriorated files is calculated. The results, shown in Table 5.3, indicate a positive trend: the number of improved files is, in both cases, greater than the number of deteriorated files. Moreover, the averaged F-measure gains are also higher than the averaged F-measure loss for both pairs.

Comparing the pairs on Table 5.3 with each other, the improvements on HH from the student-teacher learning paradigm seems to be more substantial. To further investigate the cause of this improvement, one example from the *ENST* dataset, which has the largest F-measure gain among all files, is selected. The HH activation functions generated from both teacher and student model are shown in Fig. 5.7. Compared to the teacher's activation function, the student's activation function is sharper and cleaner, demonstrating the benefit of this paradigm.

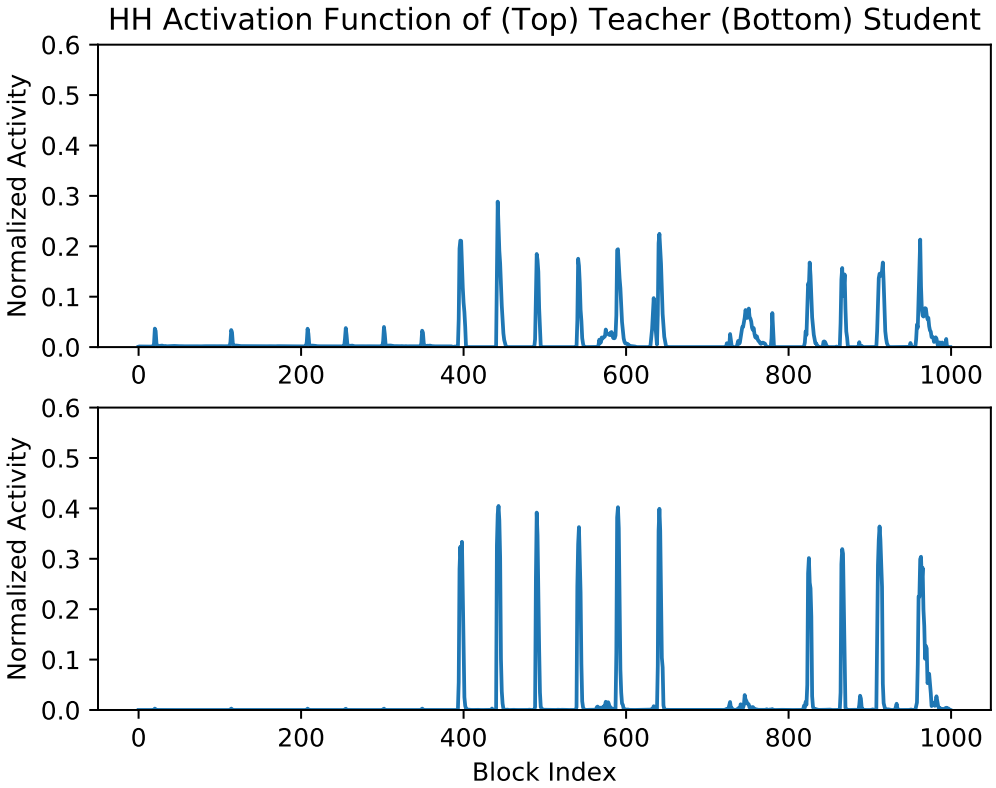


Figure 5.7: Example of the (top) teacher's and (bottom) student's HH activation function in comparison.

## 5.5 Conclusion

This chapter investigates two approaches to address the data challenge by considering both labeled and unlabeled data. First, the performance of multiple systems on all available ADT datasets is compared in an unified setting. The results indicate a potential bias when relying on one dataset and highlight the necessity of including more datasets in future ADT evaluations. Furthermore, the usefulness of unlabeled data for two major types of ADT systems via two different learning paradigms, feature learning and the student-teacher learning approach, are evaluated on multiple datasets. For both paradigms, the results are encouraging, demonstrating the potential of achieving better performance than the baseline systems on different drum instruments.

These results, while suggesting the need for additional labeled data in the field of ADT, also encourage the exploration of incorporating unlabeled data in the training. Possible directions for improving the proposed approaches are: (i) the evaluation of various methods for unsupervised feature learning such as Sparse Coding [109] and Deep Belief Networks [110], (ii) the evaluation of different combinations of teacher and student models, for example, the combination of different types of DNN either as teachers or students; the identification of suitable architectures for these roles could also be an interesting direction, and (iii) the application of outlier detection [133] approaches to filter out noisy unlabeled data.

# CHAPTER 6
# CONCLUSION

## 6.1 Summary

Labeled music data is the crucial link between machines and the human knowledge of music, and its availability will continue to be the key towards the future advancement in MIR research. Researchers have been making laudable efforts to improve the data availability by releasing new datasets or extending the existing ones. However, many research topics in MIR are still in need of more data. This is especially the case in AMT research, where the creation of annotations is both time-consuming and skill-demanding. To put this challenge in context, this thesis highlighted the data insufficiency in the field of ADT and demonstrated three possible directions for overcoming the hurdles. In the next section, the contributions and the remaining open questions regarding these directions are summarized.

## 6.2 Contributions

The main contributions of this work are the following three approaches for addressing the data challenge in ADT: (i) creating a new ADT dataset with a semi-automatic annotation process, (ii) designing an ADT system that requires minimum prior knowledge, and (iii) exploring methods that integrate unlabeled data into the training of ADT systems for potential improvement.

### 6.2.1 Dataset for ADT Tasks

The first contribution of this thesis is the creation of MDB-Drums dataset [17]. This dataset is a collaborative effort that aims to provide researchers in the field of ADT more resources for both training and testing. Furthermore, this dataset can be used as a new benchmark dataset.

Compared to the existing datasets such as IDMT-SMT-Drum [74] and ENST Drums [94], MDB-Drums contains audio recordings with diverse music genres, which better represent real-world music. To speed up the process of annotation, a semi-automatic approach is proposed to leverage an existing onset detection algorithm for automating part of the task. With careful selection of data format (e.g., multi-track files), the onset detector is able to return reliable estimation of onset times and reduces the workload of human annotators significantly. Also, the combination of automatic and manual examination ensures the quality of the resulting dataset.

Contributing a new dataset is the most straightforward solution to addressing the dataset insufficiency. However, the scalability of this approach is one of the main concerns. In particular, the need of multi-track files and the human involvement prevent this approach from being applicable to any arbitrary music data. An efficient and robust way of creating labeled data is still an open question for future ADT studies.

### 6.2.2 ADT with Limited Data

The second contribution of this thesis is the design of an ADT system that requires a minimum amount of training data [75, 76]. To this end, a signal adaptive NMF-based system with template adaptation capabilities was presented. The proposed system only requires a predefined drum dictionary as prior knowledge; this dictionary could be constructed with only a few single drum hits per instrument. Additionally, the system is designed specifically for DTM by taking the presence of melodic and pitched instruments into consideration. The inclusion of undefined harmonic dictionary and weighting matrix enables the algorithm to separate and emphasize the target drum sounds from the other instruments. To account for the variations between the drum templates and the actual drum sounds in the test signal, two template adaptation methods are proposed to iteratively update the drum dictionary. The evaluation results indicate the effectiveness of the template adaption methods for improving the precision of the system.

This system presents a simple and effective method for the DTM task, and it performs comparably with data-hungry state-of-the-art systems. One of the advantages of this system is the generalizability. Since the system does not require a large pool of training data, the evaluation can be done in the most generalized setup. The cross-dataset validation shows that the system works reasonably well across datasets. However, for the optimal performance, the open parameters have to be carefully chosen. This process relies on heuristics and domain knowledge, and the optimal strategy for automatically determining the parameters such as the harmonic rank $r_{\mathrm{H}}$ is still to be discovered.

### 6.2.3 ADT with Unlabeled Data

The third contribution of this thesis is the exploration of the usefulness of unlabeled data to general ADT systems [87]. To integrate unlabeled data to the *Classification-based* and *Activation-based* ADT systems, two learning paradigms inspired by the concepts of *feature learning* and *student-teacher learning* were implemented. These two learning paradigms provided flexible schemes for the two major types of ADT systems to take advantage of the large amount of unlabeled data. A genre balanced unlabeled dataset with real-world music is collected from online resources, and the size is considerably larger than any of the existing labeled drum datasets. The resulting systems are evaluated on multiple DTM datasets. Both paradigms show improvements on different drum instruments, suggesting a positive trend of using unlabeled data.

The proposed methods alleviate the need of labeled datasets by utilizing the large unlabeled dataset for training, and they could be used to improve the existing ADT systems in a scalable way as the unlabeled dataset can be easily extended. The remaining questions are the inconclusive results of these methods on different instruments. While the improvement for one instrument is noticeable, it is minimal for another instrument. None of the methods can improve all instruments at the same time. More experimentation and exploration of suitable models for these methods is necessary for further optimization.

### 6.2.4 Online Resources

In addition to the above mentioned contributions, this thesis also presents the following online resources as part of the contributions. These online repositories were released under the GNU general public license, and they were made available to ensure the reproducibility of this work and facilitate the future development. These repositories include:

1. **NMF Drum Toolbox**:[1] this repository consists of several NMF-based functions for drum transcription purposes. The implemented algorithms include PFNMF, AM1, AM2 (see Chapter 4), NMFD [134], and SA-NMF [74] (courtesy to Christian Dittmar). All of the functions were implemented in Matlab. This toolbox provides an easy access to the methods described in this thesis and allows comparison with other NMF-based systems.

2. **Drum PT Dataset**:[2] the extension of the existing ENST-Drums dataset [94] with the annotations of different playing techniques (e.g., flam, drag, and roll) is presented in this dataset. Specifically, the *minus one* subset was annotated for evaluating playing technique detection in the DTM setting. Only 30 out of 64 tracks contain such techniques on SD. These techniques are annotated using the snare channel of the recordings, and each technique is labeled with the starting time, duration, and the technique index. As a result, a total number of 182 events (Roll: 109, Flam: 26, Drag: 47) have been annotated, and each event has a length of approximately 250 to 400 ms.

3. **MDB-Drums Dataset**:[3] this dataset contains 23 annotated recordings for DTD and DTM tasks (see Sect. 5.3.2 for more details). The recordings were selected from the MEDLEYDB dataset [96]. This is a collaborative work that aims to provide more real-world labeled data for ADT research. All of the tracks were annotated using a semi-automatic process as shown in Fig. 3.1.

[1] https://github.com/cwu307/NmfDrumToolbox
[2] https://github.com/cwu307/DrumPtDataset
[3] https://github.com/CarlSouthall/MDBDrums

4. **ADT using Unlabeled Data**:[4] this is the repository of all the source code for conducting the experiments described in Chapter 5. This includes the implementation of data pre-processing, neural network models for both learning paradigms (e.g., feature learning and student-teacher learning), and the evaluation script. All implementations were written in Python.

5. **Unlabeled Drum Dataset**:[5] in this repository, the source code for creating the unlabeled dataset used in Chapter 5 is included. The functions, which parse the Billboard Chart and subsequently retrieve the audio files from Youtube, were written in Python. All of the Youtube links are included in the repository. However, the audio files are not provided due to copyright restrictions.

## 6.3 Future Directions

A reliable extraction of semantic information from music is the stepping stone towards the embodiment of intelligent machines that understand music, and the successful realization of this goal requires the advancement in AMT research. By focusing on the data challenge, this thesis suggests different options for advancing ADT research. To make further progress in the field of ADT, the possible directions include (but not limited to) the followings:

(i) *More data*: despite the expensive process of creating labeled data, it is still the most straightforward solution for data insufficiency. While the proposed approaches in this thesis provide alternative ways to work under the data constraints, the importance of having more data is still unquestionable. Furthermore, whether or not the existing ADT datasets can be considered as generalizable test sets is unclear. In any case, having more labeled data is beneficial for the future development of ADT systems. Since this is a labor-intensive task, contributions from the community should be highly encouraged.

[4] https://github.com/cwu307/ADT_with_unlabeledData
[5] https://github.com/cwu307/unlabeledDrumDataset

(ii) *Integration to language models*: as summarized in Table 2.1, *Language-model-based* ADT systems are still the minority. Intuitively, the model that learns the underlying vocabulary of drum sequences should provide musically meaningful transcription and thus improve the performance. However, so far these models seem to under-perform the popular *Classification-based* and *Activation-based* systems, possibly due to the lack of symbolic data for training. With the introduction of more symbolic data in the future, integrating language models to ADT systems could be a promising way of improving the performance.

(iii) *Pre-processing strategies*: most of the existing ADT systems reported a significant performance drop when switching from DTD to DTM. This implies the strong influence of melodic instruments on the systems. In this regard, a pre-processing step that suppresses these melodic instrument sounds (e.g., HPSS) could be helpful for improving the performance of the DTM task. This idea has been explored in the previous studies [19, 56], but the benefit was either marginal or inconclusive. Nevertheless, with the latest development in source separation techniques such as the contributions in Signal Separation Evaluation Campaign for Music (SiSEC MUS[6]), new strategies that are optimal for ADT tasks could be worth exploring.

Finally, with the presented work in this thesis, the author hopes to inspire more studies on new strategies for exploiting the large amount of unlabeled data in ADT research; the implications of these studies could potentially benefit other tasks in AMT, enabling exciting ways of creating more powerful MIR systems.

[6]https://www.sisec17.audiolabs-erlangen.de, last accessed 2018/04/15

# Appendices

# APPENDIX A

# COMPLETE EXPERIMENT RESULTS

The following tables contain the evaluation results of all the presented systems on each individual dataset.

Table A.1: Evaluation results of the feature-learning-paradigm-based systems on different datasets. The F-measure presented here is the average across all the tracks within each individual dataset.

| **Experiment** | | **Averaged F-measure** | | |
|---|---|---|---|---|
| **Role** | **System** | **HH** | **BD** | **SD** |
| **ENST** | | | | |
| Baseline | MFCC | 0.74 | 0.64 | 0.47 |
| Baseline | CONV-RANDOM | 0.73 | 0.59 | 0.47 |
| Evaluated | CONV-AE | 0.72 | 0.64 | 0.48 |
| Evaluated | CONV-DAE | 0.73 | 0.63 | 0.48 |
| **m2005** | | | | |
| Baseline | MFCC | 0.65 | 0.69 | 0.41 |
| Baseline | CONV-RANDOM | 0.61 | 0.57 | 0.37 |
| Evaluated | CONV-AE | 0.63 | 0.68 | 0.44 |
| Evaluated | CONV-DAE | 0.62 | 0.68 | 0.43 |
| **MDB** | | | | |
| Baseline | MFCC | 0.51 | 0.59 | 0.48 |
| Baseline | CONV-RANDOM | 0.56 | 0.50 | 0.50 |
| Evaluated | CONV-AE | 0.56 | 0.58 | 0.51 |
| Evaluated | CONV-DAE | 0.55 | 0.56 | 0.50 |
| **RBMA** | | | | |
| Baseline | MFCC | 0.55 | 0.55 | 0.25 |
| Baseline | CONV-RANDOM | 0.52 | 0.48 | 0.23 |
| Evaluated | CONV-AE | 0.51 | 0.59 | 0.26 |
| Evaluated | CONV-DAE | 0.52 | 0.58 | 0.25 |

Table A.2: Evaluation results of the student-teacher-learning-paradigm-based systems on different datasets. The F-measure presented here is the average across all the tracks within each individual dataset.

| **Experiment** | | **Averaged F-measure** | | |
|---|---|---|---|---|
| **Role** | **System** | **HH** | **BD** | **SD** |
| **ENST** | | | | |
| Teacher | PFNMF (SMT) | 0.67 | 0.78 | 0.50 |
| Teacher | PFNMF (200D) | 0.66 | 0.84 | 0.46 |
| Student | FC-200 | 0.73 | 0.67 | 0.49 |
| Student | FC-ALL | 0.72 | 0.77 | 0.51 |
| Student | FC-ALL (ALT) | 0.70 | 0.74 | 0.50 |
| **m2005** | | | | |
| Teacher | PFNMF (SMT) | 0.42 | 0.58 | 0.53 |
| Teacher | PFNMF (200D) | 0.30 | 0.64 | 0.44 |
| Student | FC-200 | 0.51 | 0.60 | 0.50 |
| Student | FC-ALL | 0.47 | 0.60 | 0.46 |
| Student | FC-ALL (ALT) | 0.51 | 0.58 | 0.51 |
| **MDB** | | | | |
| Teacher | PFNMF (SMT) | 0.44 | 0.64 | 0.43 |
| Teacher | PFNMF (200D) | 0.43 | 0.63 | 0.35 |
| Student | FC-200 | 0.48 | 0.62 | 0.45 |
| Student | FC-ALL | 0.45 | 0.61 | 0.45 |
| Student | FC-ALL (ALT) | 0.49 | 0.64 | 0.43 |
| **RBMA** | | | | |
| Teacher | PFNMF (SMT) | 0.34 | 0.43 | 0.32 |
| Teacher | PFNMF (200D) | 0.50 | 0.55 | 0.34 |
| Student | FC-200 | 0.52 | 0.38 | 0.31 |
| Student | FC-ALL | 0.48 | 0.37 | 0.27 |
| Student | FC-ALL (ALT) | 0.49 | 0.35 | 0.31 |

## REFERENCES


[1] M. Schedl, E. Gómez, and J. Urbano, "Music information retrieval: Recent developments and applications," *Foundations and Trends® in Information Retrieval*, vol. 8, no. 2-3, pp. 127–261, 2014.

[2] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri, "Automatic music transcription: Challenges and future directions," *Journal of Intelligent Information Systems*, vol. 41, no. 3, pp. 407–434, 2013.

[3] A. Klapuri, "A perceptually motivated multiple-f0 estimation method," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY, 2005, pp. 2–5.

[4] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. B. Sandler, "A tutorial on onset detection in music signals," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 13, no. 5, pp. 1–13, 2005.

[5] S. Essid, G. Richard, and B. David, "Instrument recognition in polyphonic music based on automatic taxonomies," *IEEE Transactions on Audio, Speech, and Signal Processing*, vol. 14, no. 1, 2006.

[6] F. Eyben, S. Böck, B. Schuller, and A. Graves, "Universal onset detection with bi-directional long short-term memory neural networks," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Utrecht, The Netherlands, 2010, pp. 589–594.

[7] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "CREPE: A convolutional representation for pitch estimation," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2018.

[8] S. J. Raudys and A. K. Jain, "Small sample size effects in statistical pattern recognition: Recommendations for practitioners," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 3, pp. 252–264, 1991.

[9] B. C. J. Moore, *An introduction to the psychology of hearing*. Brill, 2012.

[10] M. Goto, K. Yoshii, H. Fujihara, M. Mauch, and T. Nakano, "Songle: A Web Service for Active Music Listening Improved by User Contributions.," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2011, pp. 311–316.

[11] C. Wu and A. Lerch, "On drum playing technique detection in polyphonic mixtures," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, New York City, USA, 2016, pp. 218–224.

[12] G. Valenzise, L. Gerosa, M. Tagliasacchi, F. Antonacci, and A. Sarti, "Scream and gunshot detection and localization for audio-surveillance systems," in *Proceedgins of the IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, London, UK, 2007, pp. 21–26.

[13] A. Mesaros, T. Heittola, A. Eronen, and T. Virtanen, "Acoustic event detection in real-life recordings," in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, Aalborg, Denmark, 2010, pp. 1267–1271.

[14] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[15] A. Chaigne and J. Kergomard, *Acoustics of musical instruments*, 2nd. Springer, 2016.

[16] D. FitzGerald and J. Paulus, "Unpitched percussion transcription," in *Signal Processing Methods for Music Transcription*, A. Klapuri and M. Davy, Eds., Springer, 2006, pp. 131–162.

[17] C. Southall, C.-W. Wu, A. Lerch, and J. Hockman, "MDB DRUMS - an annotated subset of medleydb for automatic drum transcription," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)(Late-breaking Demo)*, Suzhou, China, 2017.

[18] G. L. Stone, *Stick control: For the snare drummer*. Alfred Music, 2009, ISBN: 978-1-892764-04-1.

[19] O. Gillet and G. Richard, "Transcription and separation of drum signals from polyphonic music," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 3, pp. 529–540, 2008.

[20] J. Paulus, "Signal processing methods for drum transcription and music structure analysis," PhD thesis, Tampere University of Technology, Tampere, Finland, 2009.

[21] C. W. Wu, C. Dittmar, C. Southall, R. Vogl, G. Widmer, J. Hockman, M. Mueller, and A. Lerch, "A review of automatic drum transcription," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 26, no. 9, pp. 1457–1483, 2018.

[22] A. Lerch, *An introduction to audio content analysis: Applications in signal processing and music informatics*. John Wiley & Sons, 2012.

[23] M. Müller, *Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications*, 1st. Springer, 2015.

[24] D. Fitzgerald, "Harmonic / Percussive separation using median filtering," in *Proceedings of International Conference on Digital Audio Effects (DAFx)*, 2010.

[25] W. A. Schloss, "On the automatic transcription of percussive music - from acoustic signal to high-level analysis," PhD thesis, Stanford University, 1985.

[26] A. Zils, F. Pachet, O. Delerue, and F. Gouyon, "Automatic extraction of drum tracks from polyphonic music signals," *Proceedings of the International Conference on Web delivering of Music (WEDELMUSIC)*, 2002.

[27] G. Tzanetakis, A. Kapur, and R. I. McWalter, "Subband-based drum transcription for audio signals," in *Proceedings of the Workshop on Multimedia Signal Processing*, Shanghai, China, 2005.

[28] M. A. Kaliakatsos-Papakostas, A. Floros, M. N. Vrahatis, and N. Kanellopoulos, "Real-time drums transcription with characteristic bandpass filtering," in *Proceedings of the Audio Mostly: A Conference on Interaction with Sound*, Corfu, Greece, 2012.

[29] F. Gouyon, F. Pachet, and O. Delerue, "On the use of zero-crossing rate for an application of classification of percussive sounds," in *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, Verona, Italy, 2000.

[30] P. Herrera, A. Yeterian, and F. Gouyon, "Automatic classification of drum sounds: A comparison of feature selection methods and classification techniques," in *Proceedings of the International Conference on Music and Artificial Intelligence (ICMAI)*, Edinburgh, Scotland, UK, 2002, pp. 69–80.

[31] P. Herrera, A. Dehamel, and F. Gouyon, "Automatic labeling of unpitched percussion sounds," in *Proceedings of the Audio Engineering Society Convention (AES)*, Amsterdam, Netherlands, 2003.

[32] P. Herrera, V. Sandvold, and F. Gouyon, "Percussion-related semantic descriptors of music audio files," in *Audio Engineering Society Conference: Metadata for Audio (AES)*, London, UK, 2004, pp. 69–73.

[33] V. Sandvold, F. Gouyon, and P. Herrera, "Percussion classification in polyphonic audio recordings using localized sound models," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2004, pp. 537–540.

[34] D. V. Steelant, K. Tanghe, S. Degroeve, B. D. Baets, M. Leman, J.-P. Martens, and J. P. Martens, "Classification of percussive sounds using support vector machines,"

in *Proceedings of the Annual Machine Learning Conference of Belgium and The Netherlands (BENELEARN*, 2004, pp. 146–152.

[35] A. Tindale, A. Kapur, G. Tzanetakis, and I. Fujinaga, "Retrieval of percussion gestures using timbre classification techniques," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2004.

[36] D. V. Steelant, K. Tanghe, S. Degroeve, B. D. Baets, M. Leman, and J.-P. Martens, "Support vector machines for bass and snare drum recognition," in *Classification – the Ubiquitous Challenge*, Springer, 2005, pp. 616–623.

[37] S. D. Sven, K. Tanghe, B. D. Baets, M. Leman, and J.-P. Martens, "A simulated annealing optimization of audio features for drum classification," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2005, pp. 482–487.

[38] A. Hazan, "Towards automatic transcription of expressive oral percussive performances," in *Proceedings of the International Conference on Intelligent User Interfaces (IUI)*, San Diego, California, USA, 2005, pp. 296–298.

[39] K. Tanghe, S. Degroeve, and B. D. Baets, "An algorithm for detecting and labeling drum events in polyphonic music," in *Proceedings of the Music Information Retrieval Evaluation Exchange (MIREX)*, London, UK, 2005.

[40] J. P. Bello, E. Ravelli, and M. B. Sandler, "Drum sound analysis for the manipulation of rhythm in drum loops," in *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, vol. 5, Toulouse, France, 2006.

[41] M. Miron, M. E. P. Davies, and F. Gouyon, "An open-source drum transcription system for Pure Data and Max MSP," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, BC, Canada, 2013, pp. 221–225.

[42] ——, "Improving the real-time performance of a causal audio drum transcription system," in *Proceedings of the Sound and Music Computing Conference (SMC)*, Stockholm, Sweden, 2013, pp. 402–407.

[43] M. Rossignol, M. Lagrange, G. Lafay, and E. Benetos, "Alternate level clustering for drum transcription," in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, Nice, France, 2015, pp. 2023–2027.

[44] A. Moreau and A. Flexer, "Drum transcription in polyphonic music using non-negative matrix factorisation," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Vienna, Austria, 2007, pp. 353–354.

[45] P. Roy, F. Pachet, and S. Krakowski, "Improving the classification of percussive sounds with analytical features: A case study," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Vienna, Austria, 2007, pp. 229–232.

[46] V. M. A. Souza, G. E.A.P. A. Batista, and N. E. Souza-Filho, "Automatic classification of drum sounds with indefinite pitch," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, Killarney, Ireland, 2015, pp. 1–8.

[47] N. Gajhede, O. Beck, and H. Purwins, "Convolutional neural networks with batch normalization for classifying hi-hat, snare, and bass percussion sound samples," in *Proceedings of the Audio Mostly: A Conference on Interaction with Sound*, Norrköping, Sweden, 2016, pp. 111–115.

[48] O. Gillet and G. Richard, "Automatic transcription of drum sequences using audio-visual features," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 3, Philadelphia, Pennsylvania, USA, 2005, pp. 205–208.

[49] S. Scholler and H. Purwins, "Sparse coding for drum sound classification and its use as a similarity measure," in *Proceedings of the International Workshop on Machine Learning and Music (MML)*, Florence, Italy, 2010, pp. 9–12.

[50] ——, "Sparse approximations for drum sound classification," *Journal of Selected Topics Signal Processing*, vol. 5, no. 5, pp. 933–940, 2011.

[51] L. Thompson, S. Dixon, and M. Mauch, "Drum transcription via classification of bar-level rhythmic patterns," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Taipei, Taiwan, 2014, pp. 187–192.

[52] O. Gillet and G. Richard, "Automatic transcription of drum loops," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 4, Montreal, Quebec, Canada, 2004, pp. 269–272.

[53] A. Eronen, "Musical instrument recognition using ICA-based transform of features and discriminatively trained HMMs," in *Proceedings of the International Symposium on Signal Processing and Its Applications (ISSPA)*, vol. 2, Paris, France, 2003, pp. 133–136.

[54] K. Yoshii, M. Goto, and H. G. Okuno, "Automatic drum sound description for real-world music using template adaptation and matching methods," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2004.

[55] ——, "Adamast: A drum sound recognizer based on adaptation and matching of spectrogram templates," in *Proceedings of the Music Information Retrieval Evaluation eXchange (MIREX)*, 2005.

[56] ——, "Drum sound recognition for polyphonic audio signals by adaptation and matching of spectrogram templates with harmonic structure suppression," *IEEE-transactions on audio, speech, and language processing*,

[57] T. Nakano, M. Goto, J. Ogata, and Y. Hiraga, "Voice drummer: A music notation interface of drum sounds using voice percussion input," in *Proceedings of the Annual ACM Symposium on User Interface Software and Technology (UIST)*, 2005, pp. 49–50.

[58] J. Paulus and A. Klapuri, "Drum sound detection in polyphonic music with hidden markov models," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2009, no. 14, 2009.

[59] U. Şimşekli, A. Jylhä, C. Erkut, and A. T. Cemgil, "Real-time recognition of percussive sounds by a model-based method," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2011, no. 1, 2011.

[60] G. Dzhambazov, "Towards a drum transcription system aware of bar position," in *Proceedings of the Audio Engineering Society Conference on Semantic Audio (AES)*, London, UK, 2014.

[61] O. Gillet and G. Richard, "Supervised and unsupervised sequence modelling for drum transcription," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Vienna, Austria, 2007, pp. 219–224.

[62] D. FitzGerald, B. Lawlor, and E. Coyle, "Sub-band independent subspace analysis for drum transcription," in *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, Hamburg, Germany, 2002, pp. 65–69.

[63] D. FitzGerald, R. Lawlor, and E. Coyle, "Prior subspace analysis for drum transcription," in *Proceedings of the audio engineering society convention (AES)*, 2003.

[64] D. FitzGerald, B. Lawlor, and E. Coyle, "Drum transcription in the presence of pitched instruments using prior subspace analysis," in *Proceedings of the Irish Signals and Systems Conference (ISSC)*, Limerick, Ireland, 2003.

[65] D. FitzGerald, "Automatic drum transcription and source separation," PhD thesis, Dublin Institute of Technology, Dublin, Ireland, 2004.

[66] A. Spich, M. Zanoni, A. Sarti, and S. Tubaro, "Drum music transcription using prior subspace analysis and pattern recognition," in *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, Graz, Austria, 2010.

[67] C. Dittmar and C. Uhle, "Further steps towards drum transcription of polyphonic music," in *Proceedings of the Audio Engineering Society Convention (AES)*, Berlin, Germany, 2004.

[68] J. Paulus and T. Virtanen, "Drum transcription with non-negative spectrogram factorisation," in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, Antalya, Turkey, 2005.

[69] D. S. Alves, J. Paulus, and J. Fonseca, "Drum transcription from multichannel recordings with non-negative matrix factorization," in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, Glasgow, Scotland, UK, 2009, pp. 894–898.

[70] E. Battenberg, "Techniques for machine understanding of live drum performances," PhD thesis, University of California at Berkeley, 2012.

[71] E. Battenberg, V. Huang, and D. Wessel, "Live drum separation using probabilistic spectral clustering based on the Itakura-Saito divergence," in *Proceedings of the Audio Engineering Society Conference on Time-Frequency Processing in Audio (AES)*, Helsinki, Finland, 2012.

[72] H. Lindsay-Smith, S. McDonald, and M. Sandler, "Drumkit transcription via convolutive NMF," in *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, York, UK, 2012.

[73] A. Röbel, J. Pons, M. Liuni, and M. Lagrange, "On automatic drum transcription using non-negative matrix deconvolution and Itakura Saito divergence," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brisbane, Australia, 2015, pp. 414–418.

[74] C. Dittmar and D. Gärtner, "Real-time transcription and separation of drum recordings based on NMF decomposition," in *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, Erlangen, Germany, 2014, pp. 187–194.

[75] C.-W. Wu and A. Lerch, "Drum transcription using partially fixed non-negative matrix factorization," in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, Nice, France, 2015, pp. 1281–1285.

[76] ——, "Drum transcription using partially fixed non-negative matrix factorization with template adaptation," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Malaga, Spain, 2015, pp. 257–263.

[77] E. Benetos, S. Ewert, and T. Weyde, "Automatic transcription of pitched and unpitched sounds from polyphonic music," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, 2014, pp. 3107–3111.

[78] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, vol. 9, 2010, pp. 249–256.

[79] C. Southall, R. Stables, and J. Hockman, "Automatic drum transcription using bidirectional recurrent neural networks," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, New York City, USA, 2016, pp. 591–597.

[80] R. Vogl, M. Dorfer, and P. Knees, "Recurrent neural networks for drum transcription," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, New York City, USA, 2016, pp. 730–736.

[81] ——, "Drum transcription from polyphonic music with recurrent neural networks," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, New Orleans, Louisiana, USA, 2017, pp. 201–205.

[82] C. Southall, R. Stables, and J. Hockman, "Automatic drum transcription for polyphonic recordings using soft attention mechanisms and convolutional neural networks," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Suzhou, CN, 2017, pp. 606–612.

[83] R. Vogl, M. Dorfer, G. Widmer, and P. Knees, "Drum transcription via joint beat and drum modeling using convolutional recurrent neural networks," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Suzhou, CN, 2017, pp. 150–157.

[84] T. Nakano, J. Ogata, M. Goto, and Y. Hiraga, "A drum pattern retrieval method by voice percussion," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2004, pp. 550–553.

[85] O. Gillet and G. Richard, "Drum track transcription of polyphonic music signals using noise subspace projection," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, London, UK, 2005.

[86] E. Pampalk, P. Herrera, and M. Goto, "Computational models of similarity for drum samples," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 408–423, 2008.

[87] C.-W. Wu and A. Lerch, "Automatic drum transcription using the student-teacher learning paradigm with unlabeled music data," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Suzhou, China, 2017, pp. 613–620.

[88] Y. Yang, "An evaluation of statistical approaches to text categorization," *Journal of Information Retrieval*, vol. 1, pp. 69–90, 1999.

[89] I. J. Hirsh, "Auditory perception of temporal order," *The Journal of the Acoustical Society of America*, vol. 31, no. 6, pp. 759–767, 1959.

[90] M. Prockup, E. M. Schmidt, J. Scott, and Y. E. Kim, "Toward understanding expressive percussion through content based analysis," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2013.

[91] J. Hochenbaum and A. Kapur, "Drum stroke computing: Multimodal signal processing for drum stroke identification and performance metrics," in *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, Ann Arbor, USA, 2011.

[92] *200 Drum Machines Dataset: http://www.hexawe.net/mess/200.Drum.Machines*.

[93] R. Marxer and J. Janer, "Study of regularizations and constraints in NMF-based drums monaural separation," in *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, 2013, pp. 1–6.

[94] O. Gillet and G. Richard, "ENST-drums: an extensive audio-visual database for drum signals processing," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2006.

[95] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC music database: popular, classical and jazz music databases.," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2002, pp. 287–288.

[96] R. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. Bello, "MedleyDB: a multitrack dataset for annotation-intensive MIR research," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2014.

[97] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer, "Madmom: A new python audio and music signal processing library," in *Proceedings of the ACM on Multimedia Conference*, Amsterdam, The Netherlands, 2016, pp. 1174–1178.

[98] P. Smaragdis and J. C. Brown, "Non-negative matrix factorization for polyphonic music transcription," in *Proceedings of the Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2003.

[99] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proceedings of the Advances In Neural Information Processing Systems (NIPS)*, Denver, CO, USA, 2000, pp. 556–562.

[100] P. Smaragdis, B. Raj, and M. Shashanka, "Supervised and semi-supervised separation of sounds from single-channel mixtures," in *Proceedings of the International Conference on Independent Component Analysis and Signal Separation*, 2007, pp. 414–421.

[101] S. A. Raczyski, N. Ono, and S. Sagayama, "Multipitch analysis with harmonic nonnegative matrix approximation," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2007.

[102] J. Yoo, M. Kim, K. Kang, and S. Choi, "Nonnegative matrix partial co-factorization for drum source separation," in *Proceedings of the International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2010.

[103] Jia Deng, Wei Dong, R. Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 248–255.

[104] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Master's thesis, University of Toronto, 2009.

[105] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet Classification with deep convolutional neural networks," in *Proceedings of the Advances In Neural Information Processing Systems (NIPS)*, 2012, pp. 1–9.

[106] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[107] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Transfer learning for music classification and regression tasks," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Suzhou, China, 2017.

[108] B. Mcfee, E. J. Humphrey, and J. P. Bello, "A software framework for musical data augmentation," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Malaga, Spain, 2015, pp. 248–254.

[109] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: transfer learning from unlabeled data," in *Proceedings of the International Conference on Machine Learning (ICML)*, Oregon, USA, 2007, pp. 759–766.

[110] P. Hamel and D. Eck, "Learning features from music audio with deep belief networks," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2010, pp. 339–344.

[111] M. Längkvist, L. Karlsson, and A. Loutfi, "A review of unsupervised feature learning and deep learning for time-series modeling," *Pattern Recognition Letters*, vol. 42, pp. 11–24, 2014.

[112] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, pp. 607–609, 1996.

[113] P.-K. Jao and Y.-H. Yang, "Music annotation and retrieval using unlabeled exemplars: Correlation and sparse codes," *Signal Processing Letters, IEEE*, vol. 22, no. 10, pp. 1771–1775, 2015.

[114] K. Markov and T. Matsui, "Nonnegative matrix factorization based self-taught learning with application to music genre classification," in *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing*, 2012.

[115] A. Coates, H. Lee, and A. Y. Ng, "An analysis of single-layer networks in unsupervised feature learning," in *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, Fort Lauderdale, FL, USA, 2011, pp. 215–223.

[116] N. Srivastava, E. Mansimov, and R. Salakhutdinov, "Unsupervised learning of video representations using LSTMs," in *Proceedings of the International Conference on Machine Learning (ICML)*, Lille, France, 2015.

[117] L. Su, L.-F. Yu, and Y.-H. Yang, "Sparse ceptral and phase codes for guitar playing technique classification," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Taipei, Taiwan, 2014, pp. 9–14.

[118] C.-W. Wu and A. Lerch, "Learned features for the assessment of percussive music performances," in *Proceedings of the International Conference on Semantic Computing (ICSC)*, Laguna Hills, CA, USA, 2018.

[119] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-supervised learning*. MIT Press, 2006.

[120] B. Wu, E. Zhong, D. H. Hu, A. Horner, and Q. Yang, "SMART: Semi-supervised music emotion recognition with social tagging," in *SIAM conference on Data Mining*, 2013, pp. 279–287.

[121] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," *Arxiv:1503.02531*, pp. 1–9, 2015.

[122] J. Li, R. Zhao, J. T. Huang, and Y. Gong, "Learning small-size DNN with output-distribution-based criteria," in *Proceedings of the Conference of the International Speech Communication Association (INTERSPEECH)*, 2014, pp. 1910–1914.

[123] J. Cui, B. Kingsbury, B. Ramabhadran, G. Saon, T. Sercu, K. Audhkhasi, A. Sethy, M. Nussbaum-Thom, and A. Rosenberg, "Knowledge distillation across ensembles of multiplingual models for low-resource languages," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, USA, 2017, pp. 4825–4829.

[124] S. Watanabe, T. Hori, J. L. Roux, and J. R. Hershey, "Student-teacher network learning with enhanced features," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, USA, 2017, pp. 5275–5279.

[125] C.-W. Wu and A. Lerch, "Automatic drum transcription using the student-teacher learning paradigm with unlabeled music data," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Suzhou, China, 2017, pp. 613–620.

[126] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *Arxiv:1502.03167*, pp. 1–11, 2015.

[127] D. P. Kingma and J. L. Ba, "Adam: a Method for Stochastic Optimization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015, pp. 1–15.

[128] B. Mcfee, C. Raffel, D. Liang, D. P. W. Ellis, M. Mcvicar, E. Battenberg, and O. Nieto, "Librosa: Audio and music signal analysis in python," in *Proceedings of the Python in Science Conference (SCIPY)*, 2015, pp. 18–25.

[129] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, 2012.

[130] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, X. Zheng, and G. Brain, "TensorFlow: a system for large-scale machine learning," in *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2016, pp. 265–284.

[131] C. Raffel, B. Mcfee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis, "mir_eval: A Transparent Implementation of Common MIR Metrics," *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 367–372, 2014.

[132] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the International Conference on Machine Learning (ICML)*, New York City, USA, 2008.

[133] Y.-C. Lu, C.-W. Wu, C.-T. Lu, and A. Lerch, "Automatic outlier detection in music genre datasets," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, New York City, USA, 2016, pp. 101–107.

[134] C. Dittmar and M. Müller, "Reverse Engineering the Amen Break – Score-informed Separation and Restoration applied to Drum Recordings," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 9, pp. 1531–1543, 2016.